CRCNS.org hc3 data description
Version 1.1 (July 5, 2014)

Overview.

The CRCNS.org hc3 data set contains multiunit recordings from different rat hippocampal regions while the animals were performing multiple behavioral tasks. There are 7736 cells in the data set. They were recorded from 11 animals. The number of cells recorded from each animal and brain region are shown in Table 1. Brain region EC4 indicates either entorhinal cortex layer 3 or 5 (could not be determined which); region EC? indicates in entorhinal cortex, but without layer assignment.

Table 1: Number of cells recorded. Top row: animal name. Left column: brain region.

	ec012	ec013	ec014	ec016	f01_m	g01_m	gor01	i01_m	j01_m	pin01	vvp01	total
EC2		311	180	112								603
EC3	201	362	177	116								856
EC4		276		57								333
EC5	110	416	68	154								748
EC?								82				82
Total EC	311	1365	425	439				82				2622
CA1		1185	1136	661	99	145	50	309		23	116	3724
CA3		223		646			153			45	56	1123
DG		41		94								135
Unknown		39						3	90			132
Total	311	2853	1561	1840	99	145	203	394	90	68	172	7736

Most of the recorded cells were classified as principle (pyramidal) neurons or interneurons. The number of cells classified as principle and interneuron are shown in tables 2 and 3.

Table 2: Counts of principal (pyramidal) cells. Top row: animal name. Left column: brain region.

	ec012	ec013	ec014	ec016	f01_m	g01_m	gor01	i01_m	j01_m	pin01	vvp01	total
EC2		248	146	97								491
EC3	140	239	101	88								568
EC4		214		46								260
EC5	89	300	34	128								551
EC?								51				51
Total EC	229	1001	281	359				51				1921
CA1		887	995	577	79	131	42	289		19	94	3113
CA3		217		443			138			41	43	882
DG		18		48								66
Unknown		37						1	80			118
Total	229	2160	1276	1427	79	131	180	341	80	60	137	6100

Table 3: Counts of interneurons. Top row: animal name. Left column: brain region.

	ec012	ec013	ec014	ec016	f01_m	g01_m	gor01	i01_m	j01_m	pin01	vvp01	total
EC2		45	27	13								85
EC3	37	89	66	23								215
EC4		31		8								39
EC5	16	36	20	19								91
EC?								24				24
Total EC	53	201	113	63				24				454
CA1		205	90	46	19	13	8	14		3	22	420
CA3		4		174			14			2	4	198
DG		16		36								52
Unknown		1						1	6			8
Total	53	427	203	319	19	13	22	39	6	5	26	1132

The data was obtained during 442 recording sessions. Tables giving the number of cells recorded from each region in each session are given in file *crcns-hc3-session-cell-counts.zip* which is included with the documentation for the data set. These tables are provided to assist in selecting sessions of interest based on region. However, a more powerful way of finding sessions of interest is to use database queries as described later in this document (in sections "Metadata files", "Metadata Fields" and "Using SQLite").

During each session the animal performed one of 14 behavioral tasks. The number of recording sessions and behavioral tasks used with each animal is shown in Table 4. The description of each task is shown in table 5.

Table 4: Number of recording sessions. Top row: animal name. Left column: behavioral task.

	ec012	ec013	ec014	ec016	f01_m	g01_m	gor01	i01_m	j01_m	pin01	vvp01	total
bigSquare	24	45	4	13				1	4			91
bigSquarePlus		2										2
linear	18	90	2	9								119
linearOne							3				5	8
linearTwo							3				5	8
midSquare		4	8	2								14
Mwheel	28	16	8	14	8	7		8				89
Open											3	3
plus		11										11
sleep			19	10							1	30
Tmaze							2			3	1	6
wheel		40	8	9			1					58
wheel_home				2								2
ZigZag			1									1
Total	70	208	50	59	8	7	9	9	4	3	15	442

Table 5: Behavior descriptions.

Behavior	**Description**
bigSquare	180 cm by 180 cm.
bigSquarePlus	180 cm by 180 cm, divided by plus shaped walls.
linear	linear maze, 250 cm.
linearOne	Linear maze.
LinearTwo	Linear maze.
midSquare	Square, 120 cm by 120 cm.
Mwheel	Alternation task with wheel running. See Pastalkova et al., 2008
Open	Open maze.
plus	Plus maze.
sleep	Sleeping.
Tmaze	T-maze.
wheel	Operant wheel running task, See Mizuseki et al., 2009.
wheel_home	Homecage wheel running, animal run just for fun.
Zigzag	Zigzag maze. See Royer et al., 2010 JNS.

Data files.
The data files for each recording session are stored in separate "tar.gz" files. These files are organized into top-level directories, each of which contains data for sessions recorded on the same day using the same animal and electrode placement combination. Three example top-level directories and some of the data files within them are shown in Figure 1.

```
ec012ec.11   – top level directory
|-- ec012ec.187.tar.gz   - data for individual recording session
|-- ec012ec.188.tar.gz   - "                          "
|-- ec012ec.189.tar.gz   - "                          "
ec013.53     – top level directory
|-- ec013.932.tar.gz      - data for individual recording session
|-- ec013.938.tar.gz
|-- ec013.938.mpg.tar.gz   - mpg.tar.gz –movie showing animal movements
|-- ec013.939.tar.gz
i01_maze04  – top level directory
|-- i01_maze04_MS.001.tar.gz   - data for individual recording session
|-- i01_maze04_MS.003.tar.gz
```

Figure 1: Example top-level directories and enclosed data files.

Most of the .tar.gz files (those without 'mpg' in the name) contain recorded neural data for a session. Files with suffix ".mpg.tar.gz" contain a mpg movie showing the animal movement during the corresponding session. (e.g. file "X.mpg.tar.gz" has the movie corresponding to data in "X.tar.gz").

Neural data files.
An example of the files within a session (extracted from file ec012ec.187.tar.gz) is shown below in Figure 2:

ec012ec.187.clu.1	ec012ec.187.fet.4	ec012ec.187.mm.3	ec012ec.187.spk.3
ec012ec.187.clu.2	ec012ec.187.m1m2.1	ec012ec.187.mm.4	ec012ec.187.spk.4
ec012ec.187.clu.3	ec012ec.187.m1m2.2	ec012ec.187.res.1	ec012ec.187.threshold.1
ec012ec.187.clu.4	ec012ec.187.m1m2.3	ec012ec.187.res.2	ec012ec.187.threshold.2
ec012ec.187.eeg	ec012ec.187.m1m2.4	ec012ec.187.res.3	ec012ec.187.threshold.3
ec012ec.187.fet.1	ec012ec.187.m1v	ec012ec.187.res.4	ec012ec.187.threshold.4
ec012ec.187.fet.2	ec012ec.187.mm.1	ec012ec.187.spk.1	ec012ec.187.whl
ec012ec.187.fet.3	ec012ec.187.mm.2	ec012ec.187.spk.2	ec012ec.187.xml

Figure 2: Sample data files for a recording session. Extracted from file ec012ec.187.tar.gz.

A description of these files is provided in documents "crcns-hc2-data-description" and "crcns-hc3-processing-flowchart." The first is in the hc2 data set and the second in hc3. The file formats are the same in hc2 and hc3, however, the sampling rate used for some of the data in the hc3 data sets are different. The second document (crcns-hc3-processing-flowchart) describes these differences and is probably the best description of the data files.

A fact critical for the understanding the hc2 and hc3 data sets, is that all sessions recorded from the same animal on the same day were concatenated to do spike sorting. In other words, the spike sorting results are not specific to an individual session, but apply to a group of sessions. In the hc3 data set, sessions that were merged to do spike sorting are stored within the same top level directory.

A more thorough explanation of this and the neural data files is given below.

In the experiments, data was recorded using either 4, 8, 12, or 16 electrodes (shanks). Each electrode has 8 recording sites. When possible spikes are detected (by the voltage from any recording site crossing a threshold), the time of the possible spike, and a window of data surrounding the possible spike (from all recording sites on the electrode) are stored. The time of the spike is stored in a file with extension ".res.N" and the putative spike waveforms are stored in a file with extension ".spk.N". "N" is the electrode number, which ranges from 1 to the number of electrodes. (The example in Listing 2 has four electrodes). Later, the spike waveforms are processed to generate features that can be used for spike sorting. They are stored in a file with extension .fet.N.

To do the spike sorting, all of the .fet.N files for sessions recorded on the same day using the same animal and electrode placement combination and same electrode N are concatenated and used as input to the spike sorting process. Spike sorting is done by programs KlustaKwick (https://github.com/klusta-team/klustakwik) for automatic spike sorting, then by Klusters, http://klusters.sourceforge.net/ for manual adjustment. The result of these spike sorting steps is that the spikes most likely generated by the same neuron are placed into the same category (cluster). Each cluster is assigned a cluster number, which is a non-negative integer. The cluster numbers are stored in files with extension .clu.N. Within each .clu.N file, the first line indicates the number of clusters detected in that electrode during that session, and the subsequent lines have the cluster number assigned to the sequentially detected spikes. (So, after the first line, lines in the .clu.N files have the cluster number assigned to the corresponding lines in the .res.N, .spk.N and .fet.N files).

Because the session data which are in the same top level directory are combined for spike sorting, the cluster numbers in the .clu.N files in data files within the same directory refer to the same units. Example: cluster 3 in file ec012ec.187.clu.1 (Figure 2) is associated with the same unit as cluster 3 in file ec012ec.189.clu.1; since both are from the same top level directory ("ec012ec.11", Figure 1) and thus were recorded using the same electrode placement – animal combination; and they are from the same electrode (3).

Even though the unit numbers are consistent across sessions within a top-level directory the number of units detected within sessions (number in first line of .clu.N file) may vary slightly since some units might not generate spikes during every session. For example, the first line of file:
ec012ec.187/ec012ec.187.clu.1
contains 9, indicating that 9 clusters were present in this session. However, first line of file:
ec012ec.189/ec012ec.189.clu.1
contains 8, even though that session is in the same top level directory. The reason is that the neuron for cluster number 2 was detected in in the first session, but not the second. In case it's useful for the reader, Unix commands that were used to determine this are given in Figure 3. The command lists the first line (number of clusters detected) then the count for each cluster number for both of these .clu.1 files.

```
# display cluster counts for ec012ec.187.clu.1
ec012ec.187$ head -n 1 *".clu.1"; tail -n +2 *".clu.1" | sort -n | uniq -c
9  <==  (9 clusters detected)
1 0
5595 1
   1 2   <== cluster number 2 detected once
10440 3
1002 4
6109 5
2432 6
 147 7
63421 8

# display cluster counts for ec012ec.189.clu.1
ec012ec.187$ cd ../ec012ec.189
ec012ec.189$ head -n 1 *".clu.1"; tail -n +2 *".clu.1" | sort -n | uniq -c
8 <==  (8 clusters detected)
   4 0
7329 1
   3 3   (cluster number 2, not detected in this session).
  59 4
  16 5
  97 6
 182 7
57455 8
```

Figure 3: Unix command to display cluster counts

Within each .clu.N file, cluster 0 corresponds to mechanical noise (the wave shapes do not look like neuron's spike). Cluster 1 corresponds to small, unsortable spikes. These two clusters (0 and 1) should not be used for analysis of neural data since they do not correspond to successfully sorted spikes.

The text above describes the .spk, .res, .fet and .clu files. The files with other extensions are:

.eeg – raw data, down sampled to 1.25 KHZ, in a binary format. This contains LFP signals.

.m1m2.N and .mm.N – auxiliary file generated by the Klusters program.

.m1v – movie file showing movement of animal. These can be played using the freely available "VLC media player". Not all sessions include a video file. If a video file is present, the format is either .mpg or .m1v but not both. The

.whl – file gives position of animal during the session. Extracted from the video file.

.xml – configuration file used with neuroscope software (http://neuroscope.sourceforge.net).

.threshold – generated when detecting putative spikes.

.led – Indicates timing of led synchronization light in movie. The following sessions (topdir/session) include this file: ec014.20/ec014.260, ec014.25/ec014.350, ec014.39/ec014.723, ec014.39/ec014.729, ec014.41/ec014.765, ec014.41/ec014.771, ec016.20/ec016.288

Metadata files.
There are three zip files which provide information about the hc3 data set: *crcns-hc3-channelorder.zip*, *crcns-hc3-metadata-tables.zip* and *crcns-hc3-original-docs.zip.*

channelorder.zip
The channelorder.zip file gives the relative depth in the brain of different parts of the recording electrodes for many of the sessions. A document in the hc2 data set (crcns-hc2-shank_maps.pdf) may provide this same information for some of the sessions, in an easier to understand format.

crcns-hc3-original-docs.zip
This zip file contains the original metadata files included with the data set, along with intermediate files that were derived from them during preparation of the metadata tables described in the next section. These original metadata files should NOT be used as the primary source of information about the data set. They are included because they are the most complete metadata available, even though much of the information is undocumented and potentially confusing.

crcns-hc3-metadata-tables.zip
The crcns-hc3-metadata-tables.zip file is the main metadata file in the data set and for most cases, is probably the only metadata file that will be needed. It includes information about the cells, brain regions, experiment sessions and files in the data set. This information is stored in the following tables:

cell – information about each spike sorted cell
session & session_a – information about each experimental session
epos – position of electrodes
file – information about ".tar.gz" files and any video files that are included in the data set
spike_count – has the number of spikes detected from each cell in each session

These tables are provided in CSV (comma-separated values) format, Excel format, and as tables in an SQLite database. The specific file names and contents are given below:

hc3-cell.csv – cell table CSV (coma separated value) format
hc3-session.csv & hc3-session_a.csv – sessions tables in CSV format. See note below.
hc3-epos.csv – epos table, CSV format
hc3-file.csv – file table, CSV format
hc3-spike_count.csv – spike count table, CSV format
hc3-tables.sql – table definitions for SQLite3 database
hc3-tables.db – SQLite database containing tables
hc3-tables.xlsx – Excel spreadsheet containing tables. Each table is on a separate sheet.

The fields in these tables are given in file "hc3-tables.sql" (Figure 4a and 4b). This file was used to create the tables in the SQLite database. The fields are the same in the CSV and Excel versions of the tables.

Tables session and session_a
Table "session_a" ("a" stands for "all") contains information about *all* experimental sessions (1,538) that were used to do spike sorting, including many that do not have data files provided with the data set. Table "session" has information *only* about the 442 sessions that have data files provided with the data

set. For most queries, i.e. those to find data files of interest that are in the data set, only table session should be used since using it will limit the results to only those data files included with the data set.

The table definitions for tables session_a and session are shown in Figure 4a. (In the SQLite tables, table session is actually a SQL "view" on table "session_a" restricting the view to only those sessions with data files in the data set.) The extra sessions in table session_a (the 1,096 sessions that do not have data files in the data set) consist of 1,065 which have behavior "sleep" and 30 that have a problem with the session (too short or animal did not perform the behavior or video recorder malfunction). Field "problem" in table session_a is non-empty to indicate sessions with a problem. Sessions with problems were not included in the data set, however, they were used for spike sorting and included in table spike_count. Table "session_a" was included in the data set to allow viewing the list of all sessions used to gather data from a cell and to allow comparing the firing rate of cells during sleep to the rate in other behaviors. See example 11 in section "SQLite example queries."

```
create table session_a (
  -- contains all sessions including those without data in the data set
  id integer,      -- matches row in original MatLab Beh matrix
  topdir string,   -- directory in data set containing data (tar.gz) files
  session string,  -- individual session name (corresponds to name of tar.gz file having data)
  behavior string, -- behavior, one of: Mwheel, Open, Tmaze, Zigzag, bigSquare, bigSquarePlus, circle
         -- linear, linearOne, linearTwo, midSquare, plus, sleep, wheel, wheel_home
  familiarity integer, -- number of times animal has done task, 1=animal did task for first time,
           -- 2=second time, 3=third time, 10=10 or more
  duration real,   -- recording length in seconds
  problem string   -- is empty if no problem.  Has a string ('1', '2' or 'video') if there is a problem
);


create view session
 -- contains only those sessions with data files included in hc-3 data set
as select
   s.id,       -- matches row in original MatLab Beh matrix
   s.topdir,   -- directory in data set containing data (tar.gz) files
   s.session,  -- individual session name (corresponds to name of tar.gz file having data)
   s.behavior, -- behavior, same as in table session_a
   s.familiarity, -- number of times animal has done task
   s.duration -- recording length in seconds
from session_a s, file f
where f.session = s.session
order by id;
```

Figure 4a: SQLite statements creating tables "session_a" and "session"

```
create table cell (
 id integer,        -- Id used to match original row number in MatLab PyrIntMap.Map matrix
 topdir string,     -- top level directory containing data
 animal string,     -- name of animal
 ele integer,       -- electrode number
 clu integer,       -- ID # in cluster files
 region string,     -- brain region
 nexciting integer,   -- number of cells this cell monosynaptically excited
 ninhibiting integer, -- number of cells this cell monosynaptically inhibited
 exciting integer,    -- physiologically identified exciting cells based on CCG analysis
 inhibiting integer,  -- physiologically identified inhibiting cells based on CCG analysis
       -- (Detailed method can be found in Mizuseki  Sirota Pastalkova and Buzsaki., 2009 Neuron paper.)
 excited integer,      -- based on cross-correlogram analysis, the cell is monosynaptically excited by other cells
 inhibited integer,    -- based on cross-correlogram analysis, the cell is monosynaptically inhibited by other cells
 fireRate real, -- meanISI=mean(bootstrp(100,'mean',ISI)); fireRate = SampleRate/MeanISI; ISI is interspike intervals.
 totalFireRate real,   -- num of spikes divided by total recording length for a period with a high response rate
 cellType string       -- 'p'=pyramidal, 'i'=interneuron, 'n'=not identified as pyramidal or interneuron
 eDist float,     -- "isolation distance" (see Harris, Hirase, Leinekugel, Henze and Buzsaki, Neuron, 2001)
 RefracRatio float,  -- This is an interspike interval index "R2/10" (in Fee, Mitra and Kleinfeld, Journal of
        -- Neuroscience Methods, 1996).  R2/10 = (fraction of ISI < 2ms)/(fraction of ISI < 10 ms)*9.15/1.15 (our shortest
        -- interval between spikes allowed by our spike sorting method is 0.85 ms, (10-0.85)/(2-0.85) = 9.15/1.15) ISI
 RefracViol float -- Fraction of interspike intervals less than 2 msec.
);

create table file (
  -- information about files in hc3 dataset
  topdir string,  -- directory in data set containing data (tar.gz) files
  session string, -- individual session name (corresponds to name of tar.gz file having data)
  size integer,   -- number of bytes in tar.gz file
  video_type string, -- 'mpg', 'm1v' or '-' (for no video file)
  video_size integer -- size of video file, or 0 if no video file
);

create table epos (
  -- has electrode positions for each top level directory
  -- Note, some regions do not match that in cell table.
  -- Those that differ have following meanings:
  --   DGCA3: not sure if the electrode is DG or CA3.
  --   Ctx: somewhere in the cortex (above the hippocampus)
  --   CA: somewhere in the hippocampus (do not know if it is CA1, CA3 or DG)
  topdir string,  -- directory in data set containing data (tar.gz) file
  animal string,  -- animal name
  e1 string,      -- region for electrode 1
  e2 string,      -- region for electrode 2
-- ... (e3 through e14 fields not shown)
  e16 string      -- region for electrode 16
);

create table spike_count (
-- contains number of spikes each cell has in each session (if cell could have fired).
  cellId integer,   -- id in cell table (row in original MatLab PyrIntMap.Map)
  sessId integer,   -- id in session table (row in original MatLab Beh table)
  nSpikes integer   -- number of spikes for cell in the session
);
```

Figure 4b: Create table statements for tables: cell, file, epos and spike_count. Fields for each of these tables are documented in the comments.

Metadata fields.

Some of the fields in the metadata tables are described below.

Cell quality information
All cells included in the data set satisfy a basic quality criterion. Specifically there were 7943 cells that were detected, but of those, only 7736 were included in the data set and 207 were not included because they were judged to be not good enough quality. (The "quality" of cells is specified array "Clean" in the original data file KenjiData.mat; cells with index i, for which Clean(i)=0, were not included in the data set.)

In addition to the basic quality criteria that must be satisfied for a cell to be included in the data set, three of the values in table "cell" provide information about the quality of the spike-sorted units. These are:

- *eDist* - an isolation distance (Harris, Hirase, Leinekugel, Henze and Buzsaki, Neuron, 2001).
- *RefracRatio* - an interspike interval index "R2/10 "(Fee, Mitra and Kleinfeld, Journal of Neuroscience Methods, 1996).
- *RefracViol* - fraction of interspike intervals less than 2 msec (Takehara-Nishiuchi and McNaughton, 2008).

These values were calculated using the concatenated files used for spike sorting described earlier. Due to electrode drift and the fact that different sets of cells may fire in different sessions, if individual sessions were used to calculate these values the same units would appear to have better scores than those calculated from the concatenated values. If very stringent values were used for all of these at the same time, very few cells would be classified as good. People usually use only one or two of these at once.

Having said that, the following conditions would be expected to filter for good cells:
eDist>14, RefractRatio < 0.2, RefracViol < 0.01

Firing rates
Table cell has two fields giving the cell firing rate (in spikes per second). Field "fireRate" is calculated using an average interspike interval (ISI) that is determined using the MatLab "bootstrap" function to sample from all the interspike intervals. Field "totalFireRate" is calculated as the number of spikes during a period of high response divided by the duration of the period. The time period (having high response) used to calculate totalFireRate was determined manually from the responses in the cell. Values of both of these rates will, in general, be different than an average rate calculated from the duration of sessions (given in tables session and session_a) and the number of spikes in each session (given in table spike_count). If the value of totalFireRate is close to the average rate calculated from the duration and spike counts, it suggests that the cell response is consistently high throughout all of the sessions. About 55% of the cells have a calculated average response (from tables) within about 10% of the value of totalFireRate. (See example 12 in section "SQLite example queries.")

Using SQLite.
SQLite (www.sqlite.org) is a free, open source, SQL data base engine. Precompiled binary versions are available for Linux, Mac and Windows; and are very easy to install. The SQLite database version of the data tables (file hc3-tables.db) can be used to do searches that combine information from multiple tables.

To combine information from multiple tables, tables cell, session, file and epos are related by field "topdir"; tables session and file are related by field "session" and table spike_count references the "id" field in tables cell and session. These relationships are illustrated in Figure 5. (Table session_a has the same relationships as table session and is not shown explicitly). By setting the "where" clause of an SQL statement to specify that these fields in different tables must match, the SQL select statements can be used to combine information from multiple tables.

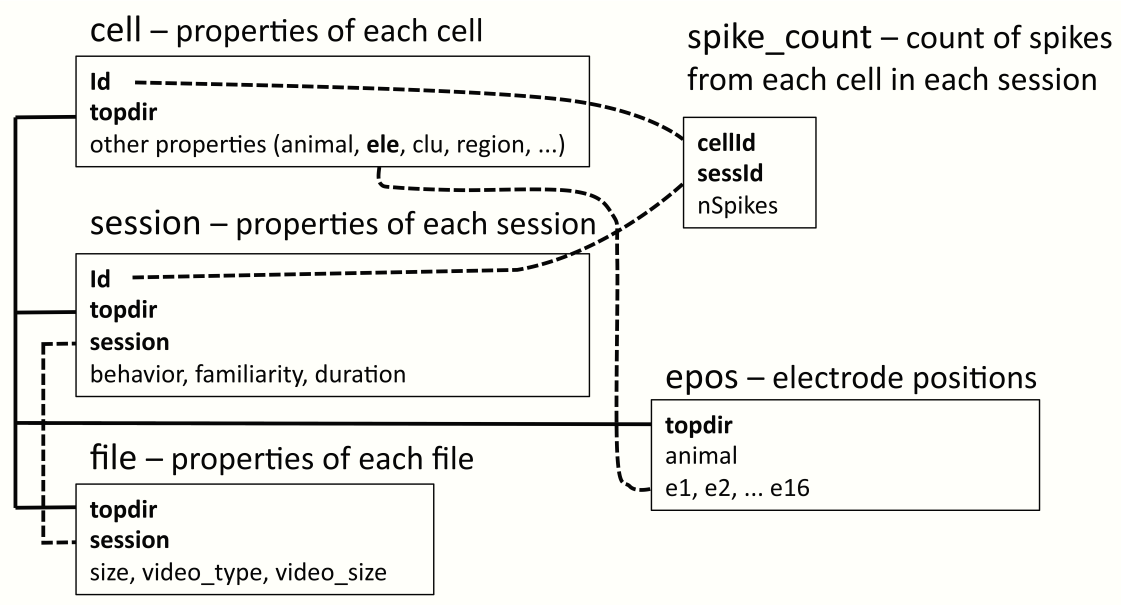

Figure 5: Relationships between tables in SQLite data base.

SQLite example queries

Several example session using SQLite to get information about the hc3 data set are given below. These examples use SQL (Structured Query Language). A tutorial for SQL sufficient to allow understanding the example and creating new queries is at: http://www.w3schools.com/sql/sql_intro.asp.

Starting SQLite

```
# Display files in directory "hc3-metadata-tables" (Unix shell)

jt$ ls
00-README.txt   hc3-epos.csv    hc3-session.csv     hc3-tables.db    hc3-tables.xlsx
hc3-cell.csv    hc3-files.csv   hc3-spike_count.csv hc3-tables.sql   old

# Start SQLite

jt$ sqlite3 hc3-tables.db
SQLite version 3.8.0.2 2013-09-03 17:11:13
Enter ".help" for instructions
Enter SQL statements terminated with a ";"

# display tables
```

```
sqlite> .tables
cell          epos          file          session       spike_count
# Display schema for table cell

sqlite> .schema cell
CREATE TABLE cell (
 id integer,       -- Id used to match original row number in MatLab PyrIntMap.Map matrix
 topdir string,    -- top level directory containing data
 animal string,    -- name of animal
 ele integer,      -- electrode number
 clu integer,      -- ID # in cluster files
 region string,    -- brain region
 nexciting integer,   -- number of cells this cell monosynaptically excited
 ninhibiting integer, -- number of cells this cell monosynaptically inhibited
 exciting integer,    -- physiologically identified exciting cells based on CCG analysis
 inhibiting integer,  -- physiologically identified inhibiting cells based on CCG analysis
      -- (Detailed method can be found in Mizuseki  Sirota Pastalkova and Buzsaki., 2009
Neuron paper.)
 excited integer,     -- based on cross-correlogram analysis, the cell is monosynaptically
excited by other cells
 inhibited integer,   -- based on cross-correlogram analysis, the cell is monosynaptically
inhibited by other cells
 fireRate real,       -- meanISI = mean(bootstrp(100,'mean',ISI)); fireRate =
SampleRate/MeanISI; ISI is interspike intervals.
 totalFireRate real,  -- num of spikes divided by total recording length
 cellType string, -- 'p'=pyramidal, 'i'=interneuron, 'n'=not identified as pyramidal or in...
 eDist float,         -- "isolation distance" (see Harris, Hirase, Leinekugel, Henze and
Buzsaki, Neuron, 2001)
 RefracRatio float,   -- This is an interspike interval index "R2/10 "
       -- (in Fee, Mitra and Kleinfeld, Journal of Neuroscience Methods, 1996)
       -- R2/10 = (fraction of ISI < 2ms)/(fraction of ISI < 10 ms)*9.15/1.15
       -- (our shortest interval between spikes allowed by our spike sorting method is
       -- 0.85 ms, (10-0.85)/(2-0.85) = 9.15/1.15). ISI
 RefracViol float  -- Fraction of interspike intervals less than 2 msec.
);
sqlite>
```

Example query 1

Displays topdir, session, animal and duration of sessions that have at least one cell in regions CA1 and EC2 and with the behavior type Square, familiarity > 5, and has a video file. Limit to first 10 records found. Order by the duration.

```
sqlite> .header on   # to display headers before results
sqlite> select distinct s.topdir, s.session, e.animal, s.duration
from cell c1, cell c2, session s, epos e, file f
where c1.topdir = c2.topdir and c1.topdir = s.topdir
and c1.topdir=e.topdir and s.session=f.session
and c1.region='CA1' and c2.region='EC2'
and (s.behavior = 'bigSquare' or s.behavior='bigSquarePlus')
and s.familiarity > 5
and f.video_type != '-'
order by s.duration desc limit 10;

topdir|session|animal|duration
ec014.24|ec014.333|ec014|5608.653  #  only one session found
sqlite>
```

Example query 2

Perform the same search as above, but this time, comment out the video file restriction.

```
sqlite> select distinct s.topdir, s.session, e.animal, s.duration
  from cell c1, cell c2, session s, epos e, file f
  where c1.topdir = c2.topdir and c1.topdir = s.topdir and
  c1.topdir=e.topdir and s.session=f.session
  and c1.region='CA1' and c2.region='EC2'
  and (s.behavior = 'bigSquare' or s.behavior='bigSquarePlus')
  and s.familiarity > 5
  -- and f.video_type != '-'  # comment out video file restriction
  order by s.duration desc limit 10;

topdir|session|animal|duration  # now 10 sessions are found
ec014.24|ec014.333|ec014|5608.653
ec013.42|ec013.754|ec013|3234.4
ec013.36|ec013.625|ec013|2568.5
ec013.33|ec013.553|ec013|2504.294
ec013.42|ec013.756|ec013|2423.603
ec013.42|ec013.757|ec013|2418.278
ec013.40|ec013.714|ec013|2278.1
ec013.40|ec013.713|ec013|2270.413
ec013.48|ec013.858|ec013|2233.958
ec013.45|ec013.808|ec013|2209.792
```

Example query 3

Finds the ten sessions with the largest number of recorded cells from region EC2 where each cell has at least 50 spikes detected. Display duration of session in minutes, average rate (spikes / sec), total number of spikes from all the cells counted. This query was used to make the tables in document *crcns-h3c-sessions-with-most-cells*. The SQLlite .separator is specified to be a tab to allow more readable output.

```
sqlite> .header on
sqlite> .separator "\t"
sqlite> select s.topdir, s.session, round(s.duration/60,1) as minutes,
round(avg(k.nSpikes / s.duration),2) as avgRate, sum(k.nSpikes) as nSpikes,
count(*) as nCells
from session s, cell c, spike_count k
where c.id = k.cellId
and s.id = k.sessId and k.nSpikes > 49 and c.region = 'EC2'
group by s.topdir, s.session
order by nCells desc, nSpikes desc limit 10;

topdir      session      minutes    avgRate   nSpikes      nCells
ec013.51    ec013.911    25.2       1.92      52167        18
ec013.51    ec013.910    21.7       2.01      47168        18
ec013.37    ec013.639    23.3       2.0       44729        16
ec013.45    ec013.805    20.5       1.14      22451        16
ec013.47    ec013.844    30.6       2.78      76456        15
ec013.51    ec013.906    21.7       1.44      28098        15
ec013.33    ec013.542    25.4       4.03      86143        14
ec013.45    ec013.808    36.8       1.4       43176        14
ec013.45    ec013.807    27.5       1.66      38411        14
ec013.45    ec013.806    24.1       1.59      32248        14
```

Example query 4

Same query as before but filtering by cell quality.

```
select s.topdir, s.session, round(s.duration/60,1) as minutes,
round(avg(k.nSpikes / s.duration),2) as avgRate, sum(k.nSpikes) as nSpikes,
count(*) as nCells
from session s, cell c, spike_count k
where c.id = k.cellId
and c.eDist>14 and c.RefracRatio < 0.2 and c.RefracViol < 0.01
and s.id = k.sessId and k.nSpikes > 49 and c.region = 'EC2'
group by s.topdir, s.session
order by nCells desc, nSpikes desc limit 10;
```

topdir	session	minutes	avgRate	nSpikes	nCells
ec014.12	ec014.123	79.9	3.05	175436	12
ec013.37	ec013.639	23.3	2.41	40472	12
ec013.51	ec013.911	25.2	1.86	33651	12
ec013.51	ec013.910	21.7	2.15	33565	12
ec013.37	ec013.643	15.3	2.62	26557	11
ec014.18	ec014.215	93.3	2.88	161080	10
ec013.37	ec013.642	32.4	2.48	48121	10
ec016.50	ec016.860	90.2	0.85	45835	10
ec013.50	ec013.898	37.0	1.39	30790	10
ec013.51	ec013.906	21.7	0.93	12178	10

Example query 5

Creates a table of number of cells detected from each region in each session, where each cell has at least 50 spikes detected. This query (without the "limit" clause) was used to make table "sess-cells" in the file "crcns-hc3-session-cell-counts.zip" by using the three SQLlite commands:

```
.mode csv
.separator ','
.output sess-cells.csv

select s.id, s.topdir, s.session,
sum(case when c.region = 'CA1' then 1 else 0 end) as CA1,
sum(case when c.region = 'CA3' then 1 else 0 end) as CA3,
sum(case when c.region = 'DG' then 1 else 0 end) as DG,
sum(case when c.region = 'EC2' then 1 else 0 end) as EC2,
sum(case when c.region = 'EC3' then 1 else 0 end) as EC3,
sum(case when c.region = 'EC4' then 1 else 0 end) as EC4,
sum(case when c.region = 'EC5' then 1 else 0 end) as EC5,
sum(case when c.region = 'EC?' then 1 else 0 end) as ECq,  -- EC?
sum(case when c.region = 'Unknown' then 1 else 0 end) as Unknown,
sum(1) as Total
from session s, spike_count k, cell c
where c.id = k.cellId and s.id = k.sessId and k.nSpikes > 49
group by s.id, s.topdir, s.session limit 10;
```

id	topdir	session	CA1	CA3	DG	EC2	EC3	EC4	EC5	ECq	Unknown	Total
5	ec012ec.11	ec012ec.187	0	0	0	0	11	0	8	0	0	19
6	ec012ec.11	ec012ec.188	0	0	0	0	8	0	7	0	0	15
7	ec012ec.11	ec012ec.189	0	0	0	0	11	0	9	0	0	20
19	ec012ec.12	ec012ec.209	0	0	0	0	3	0	4	0	0	7
20	ec012ec.12	ec012ec.210	0	0	0	0	5	0	4	0	0	9
21	ec012ec.12	ec012ec.211	0	0	0	0	5	0	4	0	0	9
22	ec012ec.12	ec012ec.212	0	0	0	0	5	0	3	0	0	8
23	ec012ec.12	ec012ec.213	0	0	0	0	5	0	2	0	0	7
35	ec012ec.13	ec012ec.227	0	0	0	0	7	0	3	0	0	10
36	ec012ec.13	ec012ec.228	0	0	0	0	9	0	4	0	0	13

Example query 6

Same query as before but filtering by cell quality. This query (without the "limit" clause) was used to make table "sess-cells-q" in the file "crcns-hc3-session-cell-counts.zip".

```
select s.id, s.topdir, s.session,
sum(case when c.region = 'CA1' then 1 else 0 end) as CA1,
sum(case when c.region = 'CA3' then 1 else 0 end) as CA3,
sum(case when c.region = 'DG' then 1 else 0 end) as DG,
sum(case when c.region = 'EC2' then 1 else 0 end) as EC2,
sum(case when c.region = 'EC3' then 1 else 0 end) as EC3,
sum(case when c.region = 'EC4' then 1 else 0 end) as EC4,
sum(case when c.region = 'EC5' then 1 else 0 end) as EC5,
sum(case when c.region = 'EC?' then 1 else 0 end) as ECq,
sum(case when c.region = 'Unknown' then 1 else 0 end) as Unknown,
sum(1) as Total
from session s, spike_count k, cell c
where c.id = k.cellId and s.id = k.sessId and k.nSpikes > 49
and c.eDist>14 and c.RefracRatio < 0.2 and c.RefracViol < 0.01
group by s.id, s.topdir, s.session limit 10;
```

id	topdir	session	CA1	CA3	DG	EC2	EC3	EC4	EC5	ECq	Unknown	Total
5	ec012ec.11	ec012ec.187	0	0	0	0	11	0	8	0	0	19
6	ec012ec.11	ec012ec.188	0	0	0	0	8	0	7	0	0	15
7	ec012ec.11	ec012ec.189	0	0	0	0	11	0	9	0	0	20
19	ec012ec.12	ec012ec.209	0	0	0	0	3	0	4	0	0	7
20	ec012ec.12	ec012ec.210	0	0	0	0	5	0	4	0	0	9
21	ec012ec.12	ec012ec.211	0	0	0	0	5	0	4	0	0	9
22	ec012ec.12	ec012ec.212	0	0	0	0	5	0	3	0	0	8
23	ec012ec.12	ec012ec.213	0	0	0	0	5	0	2	0	0	7
35	ec012ec.13	ec012ec.227	0	0	0	0	7	0	3	0	0	10
36	ec012ec.13	ec012ec.228	0	0	0	0	9	0	4	0	0	13

Example query 7

This selects sessions that have cells in both CA1 and CA3 that have at least 50 spikes detected, order by the total number of cells in CA1 and CA3.

```
select a.topdir, a.session, a.ncells as ca1Count,
   b.ncells as ca3Count,
   round(a.duration/60,1) as nMin,
   (a.nCells + b.nCells) as totalCells
from
(select
  s.topdir, s.session, s.duration, count(*) as nCells
from session s, cell c, spike_count k
where c.id = k.cellId
and s.id = k.sessId
and k.nSpikes > 49
and c.region = 'CA1'
group by s.topdir, s.session) as a
JOIN
(select
  s.topdir as topdir,
  s.session as session,
  count(*) as nCells
from session s, cell c, spike_count k
where c.id = k.cellId
and s.id = k.sessId
and k.nSpikes > 49
and c.region = 'CA3'
group by s.topdir, s.session) as b
ON
a.topdir = b.topdir
and a.session = b.session
order by totalCells desc
limit 10;
```

topdir	session	ca1Count	ca3Count	nMin	totalCells
gor01-6-7	2006-6-7_16-40-19	13	57	43.1	70
gor01-6-7	2006-6-7_11-26-53	5	62	44.9	67
pin01-11-04	11-04_22-31-40	20	28	30.2	48
pin01-11-04	11-05_0-06-51	20	25	10.4	45
gor01-6-13	2006-6-13_15-44-7	8	35	43.0	43
vvp01-4-18	2006-4-18_21-22-11	23	20	52.8	43
gor01-6-13	2006-6-13_19-11-30	6	36	47.9	42
vvp01-4-18	2006-4-18_18-26-36	24	15	26.9	39
vvp01-4-9	2006-4-9_18-43-47	36	3	82.4	39
gor01-6-12	2006-6-12_19-26-43	13	22	39.3	35

Example query 8

This selects sessions that have cells in three regions (DG, CA3 and either EC2 or EC3). Each cell must have at least 50 spikes detected. Results are ordered by the total number of cells in the three regions.

```
select a.topdir, a.session, a.ncells as nDG,
   b.ncells as nCA3, c.ncells as nEC,
   round(a.duration/60,1) as nMin,
   (a.nCells + b.nCells + c.nCells) as totalCells
from
(select
  s.topdir,
  s.session,
  s.duration,
  count(*) as nCells
from session s, cell c, spike_count k
where c.id = k.cellId
and s.id = k.sessId
and k.nSpikes > 49
and c.region = 'DG'
group by s.topdir, s.session) as a
JOIN
(select
   s.session, count(*) as nCells
from session s, cell c, spike_count k
where c.id = k.cellId
and s.id = k.sessId
and k.nSpikes > 49
and c.region = 'CA3'
group by s.session) as b
JOIN
(select
   s.session, count(*) as nCells
from session s, cell c, spike_count k
where c.id = k.cellId
and s.id = k.sessId
and k.nSpikes > 49
and c.region in ('EC2', 'EC3')
group by s.session) as c
ON
a.session = b.session
and b.session = c.session
order by totalCells desc
limit 10;
```

topdir	session	nDG	nCA3	nEC	nMin	totalCells
ec016.58	ec016.1016	19	41	9	49.7	69
ec016.57	ec016.977	13	44	8	56.0	65
ec016.58	ec016.1015	18	37	5	19.4	60
ec016.59	ec016.1035	13	40	6	71.1	59
ec016.59	ec016.1048	11	37	8	88.6	56
ec016.56	ec016.957	8	40	4	71.6	52
ec016.51	ec016.880	1	33	14	90.3	48
ec016.54	ec016.939	1	40	6	47.3	47
ec016.54	ec016.946	1	40	6	46.4	47
ec016.53	ec016.923	3	33	10	70.3	46

Example query 9

This selects the number of spikes in each unit (cluster) from session 'ec012ec.187' electrode number 1. It is the same as the first query that was done in the file system using Unix commands in Figure 3. Note that in the results, cluster 2 which appeared in results on the file system was not found in the query of the SQLlite tables. This is because that cluster (cell) was not included in the cell table because it did not pass the quality control that was done to include cells in the hc-3 data set. Specifically, there were 7943 cells that were originally detected, but of those, only 7736 were included in the hc-3 data set. The remaining 207 cells were not included because the spike sorting on them were judged to be not good enough quality. Information about these cells is still available in the original data files (array "pmMap.map" in MatLab file "KenjiData.mat"). However, using these cells for any analysis is not recommended.

```
sqlite> select nSpikes, clu from
   cell c, session s, spike_count k
   where c.id = k.cellId
   and s.id = k.sessId
   and s.session = 'ec012ec.187'
   and c.ele = 1
   order by clu;

nSpikes      clu
10440        3
1002         4
6109         5
2432         6
147          7
63421        8
sqlite>
```

Example query 10

This is the same as the previous query, but with session = 'ec012ec.189'. It is the same as the second query that was done in the file system using Unix commands in Figure 3.

```
sqlite> select nSpikes, clu from
   cell c, session s, spike_count k
   where c.id = k.cellId
   and s.id = k.sessId
   and s.session = 'ec012ec.189'
   and c.ele = 1
   order by clu;

nSpikes      clu
3            3
59           4
16           5
97           6
182          7
57455        8
```

Example query 11

This uses table "session_a" to find cells that have a different firing rate in sleep sessions than in sessions with other behaviors. Table "session_a" is the same as table session, except that in addition to information about sessions that have data files in the hc-3 data set, table "session_a" also contains information about many sleep sessions for which data was not included in the hc-3 data set. Table "session_a also has an additional field, "problem", which contains a non-empty value ('1', '2' or 'video') if there was a problem in the session.

```
select sleep.cell_id, sleep.topdir, sleep.cellType,
round(sleep.duration/3600,1) as sleep_dur,
round(other.duration/3600,1) as other_dur,
round(sleep.avg_rate,2) as sleep_rate,
round(other.avg_rate,2) as other_rate,
round((sleep.avg_rate - other.avg_rate) /
(sleep.avg_rate  + other.avg_rate), 2) as ratio
from (
-- this selects for sleep sessions
select c.id as cell_id, c.cellType, s.topdir,
sum(s.duration) as duration, sum(k.nSpikes)/sum(s.duration) as avg_rate
from cell c, session_a s, spike_count k
where c.id = k.cellId and s.id = k.sessId and s.behavior = 'sleep'
and s.problem = ''   -- needed with table session_a to not include sessions with problems
group by cell_id, c.cellType)
as sleep
JOIN
(
-- this selects for non-sleep sessions
select c.id as cell_id, c.cellType,
sum(s.duration) as duration,
sum(k.nSpikes)/sum(s.duration) as avg_rate
from cell c, session_a s, spike_count k
where c.id = k.cellId and s.id = k.sessId
and s.behavior != 'sleep'
and s.problem = ''
group by cell_id, c.cellType)
as other
on sleep.cell_id = other.cell_id
where
   sleep_dur > 1 and other_dur > 1 and  -- require that sleep & other > 1 hour
   sleep.cell_id in  -- limit to cells that had some response in all sessions
(select distinct
   cellId
from
( -- cells that average firing rate > 0.5 in every session
select
   c.id as cellId,
   min(k.nSpikes / s.duration) as min_avg_rate
from
   cell c, spike_count k, session_a s
where
   k.sessId = s.id and c.id = k.cellId and s.problem = ''
group by c.id order by c.id)
where min_avg_rate > 0.5)
order by abs(ratio) desc
limit 30;
```

Output is:

cell_id	topdir	type	sleep_dur	other_dur	sleep_rate	other_rate	ratio
5479	ec016.42	i	1.2	1.1	0.77	8.61	-0.83
2519	ec013.47	i	1.2	2.5	3.78	33.63	-0.8
7055	ec014.20	i	5.0	1.5	9.04	1.06	0.79
1822	ec013.40	n	4.8	3.5	1.47	9.73	-0.74
3988	ec014.24	i	4.5	1.6	1.68	10.56	-0.72
5921	ec016.52	i	3.8	3.4	2.59	15.87	-0.72
5970	ec016.53	i	4.0	2.3	2.98	16.17	-0.69
6023	ec016.54	i	3.7	1.6	3.06	16.33	-0.68
5585	ec016.44	i	3.6	2.7	1.32	6.41	-0.66
3519	ec014.16	p	2.9	1.1	1.17	5.32	-0.64
6754	gor01-6-7	i	2.6	1.5	8.07	35.34	-0.63
6310	ec016.59	i	4.4	3.2	2.9	12.34	-0.62
3581	ec014.16	i	2.9	1.1	8.56	35.41	-0.61
2485	ec013.46	i	3.6	2.2	9.01	36.55	-0.6
5005	ec016.27	p	3.6	2.3	2.78	0.69	0.6
5438	ec016.41	i	3.5	1.4	2.57	10.31	-0.6
2213	ec013.44	i	3.8	2.5	8.68	33.43	-0.59
3706	ec014.17	p	3.2	1.3	1.34	5.26	-0.59
5552	ec016.44	i	3.6	2.7	2.92	11.18	-0.59
6030	ec016.54	i	3.7	1.6	1.12	4.31	-0.59
245	ec012ec.22	i	1.3	1.3	5.8	22.0	-0.58
2388	ec013.45	p	4.8	2.8	1.73	6.58	-0.58
2785	ec013.49	i	4.6	3.0	8.9	32.12	-0.57
4009	ec014.24	i	4.5	1.6	4.36	15.85	-0.57
4262	ec014.29	p	5.3	1.1	3.31	0.91	0.57
5853	ec016.50	i	3.8	1.5	1.07	3.94	-0.57
6149	ec016.57	i	4.3	1.2	3.01	11.02	-0.57
1045	ec013.32	p	4.5	3.0	0.84	2.93	-0.56
3356	ec014.12	i	3.4	1.3	13.94	49.46	-0.56
3590	ec014.16	p	2.9	1.1	1.5	5.32	-0.56

In the above output, ratio is positive for cells that had average rate during sleep greater than during other behaviors, and is negative for cells that had average rate during sleep less than during other behaviors. Output is limited to only cells that have duration of at least one hour for both sleep and other behaviors and have a firing rate of at least 0.5 spikes per second for every session. If the restriction on the firing rate is not included, then the query will display mostly cells that have no response in either the sleep or other behaviors and the ratio will always be 1 or -1 (for the first thirty cells).

Example query 12

The output of the above query shows that cell with id 7055 has a stronger response during sleep than other sessions. The following query displays the average response of that cell during each individual session to show how these responses are distributed across the sessions.

```
select s.session, s.behavior, round(s.duration/3600, 2) as hours, k.nSpikes,
round(k.nSpikes/s.duration, 1) as avg_rate
from cell c, session_a s, spike_count k
where c.id = k.cellId and s.id = k.sessId
and s.problem = '' and c.id = 7055
order by session;
```

```
session         behavior      hours      nSpikes      avg_rate
ec014.254       sleep         0.59       1093         0.5
ec014.255       sleep         0.1        266          0.8
ec014.256       sleep         0.62       2159         1.0
ec014.257       sleep         0.15       490          0.9
ec014.258       sleep         0.51       2120         1.1
ec014.259       sleep         0.19       533          0.8
ec014.260       bigSquare     1.53       5812         1.1
ec014.263       sleep         0.72       37193        14.3
ec014.264       sleep         0.07       6329         26.5
ec014.265       sleep         0.41       14587        9.9
ec014.266       sleep         1.49       87401        16.3
ec014.269       sleep         0.2        11947        16.6
```

This shows that most of the responses during sleep in this cell were in sleep sessions after the non-sleep behavior, but not before.

Example query 13

In Example query 9 it was mentioned that many cells have at least one session during which there is no response (no spikes are detected). This query displays the number of cells that responded in every session for which the cell was tested, and the number of cells that had at least one session during which no spikes were detected (but could have been).

```
select nrSess, count(*) as numCells
from (
  select max(CASE WHEN k.nSpikes = 0 THEN 1 ELSE 0 END) as nrSess
  from cell c, session_a s, spike_count k
  where c.id = k.cellId and s.id = k.sessId
  and s.problem = ''   -- needed with table session_a to not include sessions with problems
  group by c.id)
group by nrSess;
```

```
nrSess          numCells
0               4330
1               3406
```

This indicated that 4,330 cells had a response in every session, and 3,406 had at least one session with no response. Total is: 7736 (number of cells in the data set).

Example query 14

This query makes a historgram count and frequency table of the percent difference between the average firing rate for each cell calculated using tables session_a and spike_count, and the average rate given by field totalFireRate in table cell. To do this, the absolute value of the difference is divided by the maximum of the two, and converted to a percentage (pdiff). This is used to display the number of cells that have at least that percent difference (count), and the percent of total cells (pfreq) with at least that difference.

```
select pdiff, count(*) as count, round(count(*)*100.0 / 7736, 1) as pfreq from
(select
   cast(abs(totalFireRate - avg_rate) /
      max(totalFireRate, avg_rate) * 10.0 as int)*10 as pdiff
   from (
      -- compute firing rate using spike_count and session table
      -- also get totalFireRate from cell table
      select c.id as cell_id, c.cellType, s.topdir, c.totalFireRate,
      sum(s.duration) as duration,
      sum(k.nSpikes)/sum(s.duration) as avg_rate
      from cell c, session_a s, spike_count k
      where c.id = k.cellId and s.id = k.sessId
      group by cell_id, c.cellType, s.topdir, c.totalFireRate
   )
) group by pdiff
order by pdiff desc;
```

pdiff	count	pfreq
100	2	0.0
90	279	3.6
80	478	6.2
70	541	7.0
60	502	6.5
50	412	5.3
40	370	4.8
30	329	4.3
20	324	4.2
10	225	2.9
0	4259	55.1
	15	0.2

Results show that two cells have totalFireRate and calculated avg_rate different by 100%, 279 are different by 90% or more (but less than 100%), 4259 have difference of less than 10%. The last row (15 cells) were not in any category because totalFireRate for these cells was not defined (have value “NaN” not a number). This is because these cells had no period in which they had a stable high firing rate.