**Global Precipitation Mission (GPM)**

**Ground Validation System**

**BULK STATISTICAL ANALYSIS PROGRAMS**

**USER’S MANUAL**

**FOR**

**GPM VALIDATION NETWORK DATA**

**February, 2017**

Goddard Space Flight Center

Greenbelt, Maryland 20771



**Document History**

| Document Version | Date | Changes |
| --- | --- | --- |
| Draft | February 2017 | Initial draft |

**Contact Information**

Additional information, including information on VN points-of-contact, can be obtained from the GPM Ground Validation web site:

http://pmm.nasa.gov/science/ground-validation

# INTRODUCTION

The GPM Validation Network (VN) Bulk Statistical Analysis Programs consist of the IDL procedures **stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match** and **z\_rain\_dsd\_profile\_scatter\_all,** which compute comparison statistics between geometry-matched spaceborne Precipitation Radar (PR) and ground radar (GR or GV) data produced by the GPM Validation Network. The spaceborne PR data are from either the TRMM Precipitation Radar (PR) or the GPM Dual-Frequency Precipitation Radar (DPR). The input data to the program consists of previously computed netCDF files containing the geometry-matched PR and GR data, one file per “rainy” site overpass (a TRMM PR or GPM DPR overpass of a GR site, with precipitation echoes present, referred to below as an “event”). These data are described in detail in the *GPM Validation Network Data User’s Guide, Vols. 1 and 2* (available at [http://pmm.nasa.gov/science/ground-validation](http://gpm.gsfc.nasa.gov/groundvalidation.html) ). For purposes of this document, the DPR designation refers to data from either the TRMM PR or GPM DPR, except where explicit differences between the two are indicated.

# Statistical Analysis of Reflectivity

The VN Statistical Analysis Program **stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match** computes mean differences (DPR-GR) between the DPR and GR reflectivity from the geo-matched data for a selected set of events, with the data stratified into vertical layers in two manners: (1) by height above the surface, in 1.5-km-deep layers, for 15 levels centered from 1.5 to 19.5 km, and (2) into three layers defined by proximity to the bright band (freezing level): above, within, and below the bright band. For purposes of the latter, match-up samples are categorized as above (below) the bright band if their base (top) is 750 m or more above (750 m or more below) the mean bright band height of the points being analyzed. The remaining points are assigned as within the bright band. Statistical results are further stratified by raincloud type (Convective, Stratiform) and by the points' distance from the radar in 3 categories: 0-49 km, 50-99 km, and (if present) 100-150 km. The proximity to the bright band (above, within, below) and the rain type (Convective, Stratiform) are merged to produce a characteristic called “Regime”, which can have seven possible values: Convective/Above BB (C\_above), Convective/Within BB (C\_in), Convective/Below BB (C\_below), Stratiform/Above BB (S\_above), Stratiform/Within BB (S\_in), Stratiform/Below BB (S\_below), and Total (all samples at a given height and distance, regardless of rain type and proximity to BB). Each stratification of data is also tagged with the site and orbit (i.e., matchup event) of the data subset. The results and their identifying metadata are written out to an ASCII, delimited text file in a format ready to be loaded into tables in the 'gpmgv' database.

The bright band heights in the netCDF file originate from the bright band analysis contained in the level 2A PR/DPR product (TRMM 2A‑25 product; GPM 2ADPR, 2AKa, or 2AKu product). An option exists to provide an external source of bright band (freezing level) heights in a file that lists the radar\_id, GPM orbit number, and model-sounding-derived freezing level. If the program determines that the netCDF matchup file for an event (site and orbit) has no valid bright band value in the data, it will go the external sounding heights file (if provided in the input parameters) and try to locate the matching model-derived freezing level as a substitute. The program can also be forced to always use the bright band substitute from the external file, for instance, to make sure that comparisons from two different DPR versions use the same bright band height value to minimize the differences in bright band proximity categorization between the two versions when the DPR bright band algorithm changes.

By default, the attenuation-corrected DPR reflectivity from the level 2A products are used in the reflectivity comparisons, even though the “raw calibrated” DPR reflectivity also is present in the netCDF data files. The procedure provides an option to compare the “raw” DPR reflectivity to the GR.

The primary output of the **stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match** program is the ASCII-delimited file of statistics stratified by site, orbit, height, merged rain type/BB proximity (regime), and range category (0=0-49 km, 1=50-99 km). The following shows an example of the selected statistics from one event (GR site and GPM orbit), i.e., as computed from one matchup netCDF file:

radar\_id | orbit | rangecat | regime | height | meandiff | diffstddev | numpts

----------+-------+----------+---------+--------+----------+------------+--------

KCRP | 2425 | 0 | C\_below | 1.5 | -1.00954 | 1.38244 | 6

KCRP | 2425 | 0 | C\_in | 1.5 | 1.62314 | 2.13489 | 4

KCRP | 2425 | 0 | S\_in | 1.5 | 2.49595 | -99.999 | 1

KCRP | 2425 | 0 | Total | 1.5 | 0.26648 | 2.12911 | 11

KCRP | 2425 | 0 | C\_in | 3 | 1.70547 | 3.09848 | 16

KCRP | 2425 | 0 | S\_in | 3 | 1.46641 | 1.06262 | 5

KCRP | 2425 | 0 | Total | 3 | 1.64855 | 2.72712 | 21

KCRP | 2425 | 0 | C\_above | 4.5 | 3.38221 | 2.50918 | 14

KCRP | 2425 | 0 | C\_in | 4.5 | 2.4858 | 0.72515 | 3

KCRP | 2425 | 0 | S\_above | 4.5 | -2.27472 | 1.32034 | 3

KCRP | 2425 | 0 | Total | 4.5 | 2.39921 | 2.95113 | 20

KCRP | 2425 | 0 | C\_above | 6 | 3.75998 | 1.64193 | 10

KCRP | 2425 | 0 | S\_above | 6 | 1.76002 | -99.999 | 1

KCRP | 2425 | 0 | Total | 6 | 3.26294 | 1.93098 | 12

KCRP | 2425 | 0 | C\_above | 7.5 | 0.93467 | 3.41505 | 3

KCRP | 2425 | 0 | Total | 7.5 | 0.78851 | 2.80366 | 4

KCRP | 2425 | 0 | C\_above | 9 | 2.8266 | 1.53323 | 3

KCRP | 2425 | 0 | Total | 9 | 2.8266 | 1.53323 | 3

KCRP | 2425 | 0 | C\_above | 10.5 | 0.41937 | 2.42242 | 3

KCRP | 2425 | 0 | Total | 10.5 | 0.41937 | 2.42242 | 3

KCRP | 2425 | 1 | C\_below | 1.5 | -1.20461 | 1.87937 | 10

KCRP | 2425 | 1 | C\_in | 1.5 | 0.20364 | 1.38004 | 9

KCRP | 2425 | 1 | S\_below | 1.5 | 1.5938 | 0.88425 | 3

KCRP | 2425 | 1 | S\_in | 1.5 | 1.52237 | -99.999 | 1

KCRP | 2425 | 1 | Total | 1.5 | -0.16998 | 1.81841 | 23

KCRP | 2425 | 1 | C\_in | 3 | 0.87192 | 1.36256 | 15

KCRP | 2425 | 1 | S\_in | 3 | 1.84306 | 0.78742 | 7

KCRP | 2425 | 1 | Total | 3 | 1.18092 | 1.27641 | 22

KCRP | 2425 | 1 | C\_above | 4.5 | 3.2473 | 3.31078 | 5

KCRP | 2425 | 1 | C\_in | 4.5 | 2.92705 | -99.999 | 1

KCRP | 2425 | 1 | S\_above | 4.5 | 1.91936 | 0.83821 | 4

KCRP | 2425 | 1 | S\_in | 4.5 | 1.1953 | 1.75712 | 2

KCRP | 2425 | 1 | Total | 4.5 | 2.43597 | 2.27074 | 12

KCRP | 2425 | 1 | C\_above | 6 | 4.56675 | 1.98774 | 4

KCRP | 2425 | 1 | Total | 6 | 4.56675 | 1.98774 | 4

KCRP | 2425 | 1 | C\_above | 7.5 | 3.78137 | -99.999 | 1

KCRP | 2425 | 1 | Total | 7.5 | 2.65756 | 1.5893 | 2

KCRP | 2425 | 1 | C\_above | 9 | 2.64473 | -99.999 | 1

KCRP | 2425 | 1 | Total | 9 | 2.64473 | -99.999 | 1

As seen in the first line of output, there are 6 samples in the 0-49 km range category, in the 1.5 km layer, with Convective rain type, lying below the bright band. The mean difference between DPR and GR reflectivity for these 6 samples is -1.01 dBZ, and the standard deviation of the reflectivity differences for these 6 samples is 1.38 dBZ. For a given matchup file, the program has iterated through all defined height layers, regimes, and range categories, found all the samples having this combination of attributes, and computed the difference statistics for these samples and writes them to a line of the output ASCII file with their identifying metadata. It repeats this for each matchup file meeting the file selection criteria, appending the output statistics to the ASCII file.

The procedure **stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match** takes a number of keyword parameters that control the set of data files to include in the processing, filtering of samples based on quality and other criteria, adjustments to GR or DPR reflectivity, and the location and type of outputs from the program. Each parameter is described in detail in the prologue of the source code file **stats\_by\_dist/stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match.pro**. They are summarized by type in the following table:

|  |  |
| --- | --- |
| MATCHUP FILES TO INCLUDE IN PROCESSING | MATCHUP\_TYPE = matchup\_type |
| KUKA = KuKa |
| SCANTYPE = swath |
| NCSITEPATH = ncsitepath |
| FILEPATTERN = filepattern |
| SITELIST = sitelist |
| EXCLUDE = exclude |
| FIRST\_ORBIT = first\_orbit |
| VERSION2MATCH = version2match |
| DATA FILTERING AND ADJUSTMENTS | PCT\_ABV\_THRESH = pctAbvThresh |
| RAY\_RANGE = ray\_range |
| MAX\_BLOCKAGE = max\_blockage\_in |
| GV\_CONVECTIVE = gv\_convective |
| GV\_STRATIFORM = gv\_stratiform |
| S2KU = s2ku |
| ALTFIELD = altfield |
| BB\_RELATIVE = bb\_relative |
| ALT\_BB\_FILE = alt\_bb\_file |
| FORCEBB = forcebb |
| DPR\_Z\_ADJUST = dpr\_z\_adjust |
| GR\_Z\_ADJUST = gr\_z\_adjust |
| ET\_RANGE = et\_range |
| DO\_STDDEV = do\_stddev |
| PROGRAM OUTPUTS | NAME\_ADD = name\_add |
| OUTPATH = outpath |
| PROFILE\_SAVE = profile\_save |
| SCATTERPLOT = scatterplot |
| BINS4SCAT = bins4scat |
| PLOT\_OBJ\_ARRAY = plot\_obj\_array |
| CONVBELOWSCAT = convbelowscat |

The data in the output ASCII file must be loaded into compatible tables in the PostgreSQL database ‘gpmgv’ so that various types of summary statistics can be aggregated and computed. The following is a sequence of IDL runs and matching PostgreSQL sessions to run and load statistics to the database for further analysis.

Run **stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match** in IDL to produce an output file to be loaded to the database. We will process PPS version V04A data files for matchup file version 1\_21 (1.21 in the code, 1\_21 in the filename and directory paths), from 2ADPR NS scan matchups, limiting the files to those whose basenames match the pattern 'GRtoDPR.K\*.15dbzGRDPR\_newDm.\*'. and having a corresponding matchup file based on the PPS v5 test run ‘ITE109’ (VERSION2MATCH='ITE109'). We will leave the GR reflectivity unadjusted (s2ku=0), will filter out any points that have less that 100% of the original GR and DPR range gates in the average below the Z thresholds used in the matchups (pct=100), and will exclude samples that have more than 10% beam blockage for the GR (MAX\_BLOCKAGE=0.1). Note that some of the IDL keywords have been truncated as IDL allows (e.g., PCT instead of PCT\_ABV\_THRESH). From the IDL prompt:

IDL> stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match, PCT=100, $

NCSITE='/data/gpmgv/netcdf/geo\_match/GPM/2ADPR/NS/V04A/1\_21/', $

FILEPATTERN='GRtoDPR.K\*.15dbzGRDPR\_newDm.\*', NAME\_ADD='Block10\_V04A', $

out='/tmp', alt\_BB='/data/tmp/GPM\_rain\_event\_bb\_km\_Uniq.txt', $

VERSION2MATCH='ITE109', MAX\_BLOCKAGE=0.1, s2ku=0

IDL should run and produce the output file:

/tmp/StatsByDist\_DPR\_GR\_Pct100\_Block10\_V04A\_AltBB\_DefaultS.unl

IDL formats this output filename and path based on control parameter values (OUTPATH, PCT\_ABV\_THRESH, ALT\_BB\_FILE, S2KU, NAME\_ADD) and the type of matchup file being processed (GRtoDPR). Exit IDL and start a PostgreSQL session in the ‘gpmgv’ database with the ‘psql’ utility:

=> psql gpmgv

psql (8.4.20)

Type "help" for help.

gpmgv=>

At the gpmgv=> prompt, enter and run the multi-line SQL command to create a table into which to load the .unl file:

create table zdiff\_stats\_dpr\_v4 (

percent\_of\_bins int,

rangecat int,

gvtype char(4),

regime character varying(10),

radar\_id character varying(15),

orbit integer,

height float,

meandiff float,

diffstddev float,

prstddev float,

gvstddev float,

prmean float,

gvmean float,

gvbinmax float,

gvbinstddev float,

numpts int,

primary key (percent\_of\_bins, rangecat, gvtype, regime, radar\_id, orbit, height)

);

Note that the ‘;’ is the termination of regular SQL commands to instruct psql that the current command is complete.

If this table already exists in the ‘gpmgv’ database an error to that effect will be reported. No harm, no foul. Next, clear the contents of the table just to make sure we aren’t working with data from an old file loading. Still within psql, run the command:

delete from zdiff\_stats\_dpr\_v4;

Now load the data from the file to the table zdiff\_stats\_dpr\_v4. Still within psql, run the SQL command:

\copy zdiff\_stats\_dpr\_v4 from '/tmp/StatsByDist\_DPR\_GR\_Pct100\_Block10\_V04A\_AltBB\_DefaultS.unl' with delimiter '|'

Note that unlike the preceding SQL commands, the \copy command is a meta-command and does not need to be terminated with the semicolon character. Although it is shown on several lines above, in practice it needs to entered as a single line with no newlines interspersed. Like in IDL, the filename string must be enclosed in quotes. The **with delimiter ‘|’** indicates to PostgreSQL that the individual fields in the data file are delimited by the vertical bar character, and must be included to prevent errors in loading. The data fields in the .unl file must match in number and type to the columns defined in the database table. The combination of fields in the PRIMARY KEY definition must be unique among the lines of data contained in the .unl file or the data will not load to the table. In normal circumstances this should not occur.

Exit the ‘psql’ session by typing the **\q** command at the prompt:

gpmgv=> \q

[morris@ds1-gpmgv ~]$

This completes the running of **stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match** and the loading of its output to the data table in the ‘gpmgv’ database. In practice, we will want to make other runs of **stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match** so that we can compare one version’s statistics to another’s. So let us run the program again in IDL with the same basic parameters, but processing the v5 test run’s data in PPS version ‘ITE109’ to the current baseline data version ‘V04A’. At the IDL prompt:

IDL> stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match, PCT=100, $

NCSITE='/data/gpmgv/netcdf/geo\_match/GPM/2ADPR/NS/ITE109/1\_21/', $

FILEPATTERN='GRtoDPR.K\*.15dbzGRDPR\_newDm.\*', NAME\_ADD='Block10\_ITE109', $

out='/tmp', alt\_BB='/data/tmp/GPM\_rain\_event\_bb\_km\_Uniq.txt', $

VERSION2MATCH='V04A’, MAX\_BLOCKAGE=0.1, s2ku=0

IDL should run and produce the output file:

/tmp/StatsByDist\_DPR\_GR\_Pct100\_Block10\_ITE109\_AltBB\_DefaultS.unl

Exit IDL, start a psql session as above, and define a new table to be loaded with the ITE109 statistics. Enter and run the SQL commands from the gpmgv=> prompt:

create table zdiff\_stats\_dpr\_v5 (

percent\_of\_bins int,

rangecat int,

gvtype char(4),

regime character varying(10),

radar\_id character varying(15),

orbit integer,

height float,

meandiff float,

diffstddev float,

prstddev float,

gvstddev float,

prmean float,

gvmean float,

gvbinmax float,

gvbinstddev float,

numpts int,

primary key (percent\_of\_bins, rangecat, gvtype, regime, radar\_id, orbit, height)

);

delete from zdiff\_stats\_dpr\_v5;

\copy zdiff\_stats\_dpr\_v5 from '/tmp/StatsByDist\_DPR\_GR\_Pct100\_Block10\_ITE109\_AltBB\_DefaultS.unl' with delimiter '|'

Exit the ‘psql’ session by typing the **\q** command at the prompt:

gpmgv=> \q

[morris@ds1-gpmgv ~]$

Now let’s make two more runs of IDL exactly as we’ve done above, but with the S2KU parameter set to 1 so that the GR S-band reflectivity is frequency-adjusted to the DPR Ku-band equivalent. First, process the V04A data with matching ITE109 events:

IDL> stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match, PCT=100, $

NCSITE='/data/gpmgv/netcdf/geo\_match/GPM/2ADPR/NS/V04A/1\_21/', $

FILEPATTERN='GRtoDPR.K\*.15dbzGRDPR\_newDm.\*', NAME\_ADD='Block10\_V04A', $

out='/tmp', alt\_BB='/data/tmp/GPM\_rain\_event\_bb\_km\_Uniq.txt', $

VERSION2MATCH='ITE109', MAX\_BLOCKAGE=0.1, s2ku=1

IDL should run and produce the output file:

/tmp/StatsByDist\_DPR\_GR\_Pct100\_Block10\_V04A\_AltBB\_S2Ku.unl

Next, process the ITE109 data with matching V04A events:

At the IDL prompt:

IDL> stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match, PCT=100, $

NCSITE='/data/gpmgv/netcdf/geo\_match/GPM/2ADPR/NS/ITE109/1\_21/', $

FILEPATTERN='GRtoDPR.K\*.15dbzGRDPR\_newDm.\*', NAME\_ADD='Block10\_ITE109', $

out='/tmp', alt\_BB='/data/tmp/GPM\_rain\_event\_bb\_km\_Uniq.txt', $

VERSION2MATCH='V04A’, MAX\_BLOCKAGE=0.1, s2ku=0

IDL should run and produce the output file:

/tmp/StatsByDist\_DPR\_GR\_Pct100\_Block10\_ITE109\_AltBB\_S2Ku.unl

Exit IDL, start a psql session as above, and define two new tables to be loaded with the Ku-adjusted V04A and ITE109 statistics, respectively. First, load the V04A table. Enter and run the SQL commands from the gpmgv=> prompt:

create table zdiff\_stats\_dpr\_v4\_s2ku (

percent\_of\_bins int,

rangecat int,

gvtype char(4),

regime character varying(10),

radar\_id character varying(15),

orbit integer,

height float,

meandiff float,

diffstddev float,

prstddev float,

gvstddev float,

prmean float,

gvmean float,

gvbinmax float,

gvbinstddev float,

numpts int,

primary key (percent\_of\_bins, rangecat, gvtype, regime, radar\_id, orbit, height)

);

delete from zdiff\_stats\_dpr\_v4\_s2ku;

\copy zdiff\_stats\_dpr\_v4\_s2ku from '/tmp/StatsByDist\_DPR\_GR\_Pct100\_Block10\_V04A\_AltBB\_S2Ku.unl' with delimiter '|'

Next, load the ITE109 (v5) table. Enter and run the SQL commands from the gpmgv=> prompt:

create table zdiff\_stats\_dpr\_v5 (

percent\_of\_bins int,

rangecat int,

gvtype char(4),

regime character varying(10),

radar\_id character varying(15),

orbit integer,

height float,

meandiff float,

diffstddev float,

prstddev float,

gvstddev float,

prmean float,

gvmean float,

gvbinmax float,

gvbinstddev float,

numpts int,

primary key (percent\_of\_bins, rangecat, gvtype, regime, radar\_id, orbit, height)

);

delete from zdiff\_stats\_dpr\_v5;

\copy zdiff\_stats\_dpr\_v5 from '/tmp/StatsByDist\_DPR\_GR\_Pct100\_Block10\_ITE109\_AltBB\_DefaultS.unl' with delimiter '|'

Now that we have loaded multiple runs of statistics into multiple tables, there are queries we can run in psql to produce summary statistics on their own, or as input to other IDL programs that can display the data as site bias maps or site-specific time series plots. Note that in the queries that follow that store intermediate results in temporary database tables, these temp tables are automatically deleted when the current psql session is exited.

First, for the unadjusted GR data tables, we will merge the DPR-GR difference statistics in the two versions into a single temporary database table, matched one-to-one by primary key attributes. From the psql session:

select a.percent\_of\_bins, a.rangecat, a.gvtype, a.regime, a.radar\_id, a.orbit, a.height, a.meandiff as meandiffv4, b.meandiff as meandiffv5, a.numpts as numptsv4, b.numpts as numptsv5 into temp merged\_diffs from zdiff\_stats\_dpr\_v4 a join zdiff\_stats\_dpr\_v5 b USING (percent\_of\_bins, rangecat, gvtype, regime, radar\_id, orbit, height);

Run a query to see how many rows of data are in the merged temp table:

select count(\*) from merged\_diffs;

Now let’s merge the two versions’ data for the Ku-adjusted statistics:

select a.percent\_of\_bins, a.rangecat, a.gvtype, a.regime, a.radar\_id, a.orbit, a.height, a.meandiff as meandiffv4, b.meandiff as meandiffv5, a.numpts as numptsv4, b.numpts as numptsv5 into temp merged\_diffs\_s2ku from zdiff\_stats\_dpr\_v4\_s2ku a join zdiff\_stats\_dpr\_v5\_s2ku b USING (percent\_of\_bins, rangecat, gvtype, regime, radar\_id, orbit, height);

Run a query to see how many rows of data are in the Ku-adjusted merged temp table:

select count(\*) from merged\_diffs\_s2ku;

Now run a query against a merged data table to compute the mean site-specific reflectivity biases between the DPR and the GR for both PPS versions from the Ku-adjusted data, aggregated over the entire dataset. Let’s limit the output to only those radars whose names fall between 'KFTG' and 'KMOB':

gpmgv=> select radar\_id, round((sum(meandiffv5\*numptsv5)/sum(numptsv5))\*100)/100\*(-1) as meandiff\_v5, sum(numptsv5) as total\_v5, round((sum(meandiffv4\*numptsv4)/sum(numptsv4))\*100)/100\*(-1) as meandiff\_v4, sum(numptsv4) as total\_v4, f.latitude, f.longitude from merged\_diffs\_s2ku a, fixed\_instrument\_location f where regime='S\_above' and numptsv5>5 and numptsv4>5 and percent\_of\_bins=100 and a.radar\_id=f.instrument\_id and a.radar\_id between 'KFTG' and 'KMOB' group by radar\_id, latitude,longitude order by radar\_id;

radar\_id | meandiff\_v5 | total\_v5 | meandiff\_v4 | total\_v4 | latitude | longitude

----------+-------------+----------+-------------+----------+----------+-----------

KFTG | -1.45 | 179 | -0.29 | 200 | 39.7867 | -104.546

KFWS | -3.65 | 223 | -2.25 | 187 | 32.5731 | -97.3031

KGRK | -2.14 | 672 | -1.18 | 635 | 30.7219 | -97.3831

KGRR | -0.52 | 693 | 0.65 | 563 | 42.8939 | -85.5447

KHGX | -2.08 | 222 | -0.32 | 98 | 29.4719 | -95.0792

KHTX | -1.63 | 278 | -0.42 | 264 | 34.9306 | -86.0833

KICT | -1.07 | 415 | 0.41 | 418 | 37.6547 | -97.4428

KILN | -1.53 | 178 | -0.36 | 67 | 39.4203 | -83.8217

KILX | -2.11 | 265 | -1.06 | 266 | 40.1506 | -89.3369

KINX | -1.34 | 117 | 0.13 | 125 | 36.175 | -95.5647

KIWX | -2.22 | 70 | -1.04 | 66 | 41.4086 | -85.7

KJAX | -1.57 | 234 | -0.37 | 232 | 30.4847 | -81.7019

KJGX | -3.1 | 32 | -1.85 | 38 | 32.6753 | -83.3511

KJKL | 0.02 | 80 | 0.76 | 102 | 37.5908 | -83.3131

KLCH | 0.23 | 1245 | 1.44 | 1327 | 30.1253 | -93.2158

KLGX | -1.1 | 471 | 0.1 | 472 | 47.1158 | -124.107

KLIX | -2.78 | 432 | -1.43 | 292 | 30.3367 | -89.8256

KLOT | -1.29 | 200 | -0.15 | 194 | 41.6047 | -88.0847

KLSX | -1.79 | 373 | -0.92 | 277 | 38.6989 | -90.6828

KLZK | -2.46 | 255 | -1.33 | 230 | 34.8364 | -92.2622

KMHX | -1.13 | 478 | 0.01 | 466 | 34.7761 | -76.8761

KMKX | 0.83 | 110 | 1.43 | 87 | 42.9678 | -88.5506

KMOB | -1.57 | 477 | -0.39 | 463 | 30.6794 | -88.2397

(23 rows)

## Producing Input Data for Bias Map Plots

As it turns out, the preceding table of data is exactly the output needed to run the (poorly named) IDL procedure **agu\_nexrad\_bias\_maps\_dpr\_v3\_v4**, which plots the mean site difference for each version on a CONUS map background, with plotted, filled circles at each radar location color coded by their site bias values. This program wants the data in the table above, but without the header lines and without the whitespace that aligns the values. So we run the query again with a few meta-options to specify the format and location of the output file, and with the site criteria modified to include all the CONUS WSR-88D sites in the VN dataset. The output will be directed to an ASCII text file **/tmp/GR\_DPR\_Bias\_KuAdj\_LL\_Pct100.unl**:

gpmgv=> \t \a \o /tmp/GR\_DPR\_Bias\_KuAdj\_LL\_Pct100.unl \\

select radar\_id, round((sum(meandiffv5\*numptsv5)/sum(numptsv5))\*100)/100\*(-1) as meandiff\_v5, sum(numptsv5) as total\_v5, round((sum(meandiffv4\*numptsv4)/sum(numptsv4))\*100)/100\*(-1) as meandiff\_v4, sum(numptsv4) as total\_v4, f.latitude, f.longitude from merged\_diffs\_s2ku a, fixed\_instrument\_location f where regime='S\_above' and numptsv5>5 and numptsv4>5 and percent\_of\_bins=100 and a.radar\_id=f.instrument\_id and a.radar\_id between 'KAAA' and 'KTYX' group by radar\_id, latitude,longitude order by radar\_id;

Note that we set the tuples\_only (\t) and align (\a) meta-options to format the output. Tuples only suppresses the header lines of the output. The align option eliminates the whitespace that aligns the data as a human-readable table. Note that these are 'toggle' options. Once they are set they remain in place for the ***psql*** session until set again to reverse the option in effect. Also, we specified that the query output should be directed to a named file rather than to the terminal (via the **\o /tmp/GR\_DPR\_Bias\_KuAdj\_LL\_Pct100.unl** option). Note that unless a different file name is specified for a subsequent query, its output will ALSO be sent to this file and appended to the end of it. Here is a partial listing of the contents of the output file, listing some of the same stations shown in the formatted table:

KFTG|-1.45|179|-0.29|200|39.7867|-104.546

KFWS|-3.65|223|-2.25|187|32.5731|-97.3031

KGRK|-2.14|672|-1.18|635|30.7219|-97.3831

KGRR|-0.52|693|0.65|563|42.8939|-85.5447

KHGX|-2.08|222|-0.32|98|29.4719|-95.0792

KHTX|-1.63|278|-0.42|264|34.9306|-86.0833

KICT|-1.07|415|0.41|418|37.6547|-97.4428

KILN|-1.53|178|-0.36|67|39.4203|-83.8217

KILX|-2.11|265|-1.06|266|40.1506|-89.3369

KINX|-1.34|117|0.13|125|36.175|-95.5647

KIWX|-2.22|70|-1.04|66|41.4086|-85.7

KJAX|-1.57|234|-0.37|232|30.4847|-81.7019

You can see that the header and whitespace have been suppressed in the data in the file, and it is in a simple format to allow IDL’s STRSPLIT function to split the rows of data into individual components based on the ‘|’ delimiter (the default delimiter for data written to an output file by ***psql***). Since there is no header information in the file, the IDL program must understand the meaning of each field in the output file by position, so the query is directly linked to how the IDL plotting program **agu\_nexrad\_bias\_maps\_dpr\_v3\_v4** will use the data. So let’s exit from the ***psql*** session and run the IDL program to plot the maps to the screen (poor quality) and to a Postscript file (higher quality).

IDL> agu\_nexrad\_bias\_maps\_dpr\_v3\_v4, '/tmp/GR\_DPR\_Bias\_KuAdj\_LL\_Pct100.unl'

As IDL runs and plots the various bias maps it will pause at each map plot and prompt the user whether to proceed or quit:

Hit Return to do next plot, Q to Quit and skip Postscript/PDF output:

After the last map has been plotted to the screen the IDL program will then write a message indicating that the maps are being plotted to a Postscript file:

\*\*\*\*mapping to file named /data/tmp/map\_ITE109\_V4\_DPR\_bias.ps

Once that is complete the program will print messages indicating that two output files have been produced, and Postscript (.ps) file and this file converted to an Adobe PDF (.pdf) file:

Finished making hard-copy map. See graphics file(s):

/data/tmp/map\_ITE109\_V4\_DPR\_bias.pdf

/data/tmp/map\_ITE109\_V4\_DPR\_bias.ps

IDL>

Note that the name of the Postscript output file (outfile) is hard-coded in the IDL program:

**outfile = '/data/tmp/map\_ITE109\_V4\_DPR\_bias'+nameAdd+'.ps'**

and the pathname of the corresponding output PDF file (outfilePDF) is derived from ‘outfile’, where the ‘.ps’ specification is replaced by ‘.pdf:

**outfilePDF='/data/tmp/map\_ITE109\_V4\_DPR\_bias'+nameAdd+'.pdf'**

As IDL just uses a ‘spawn’ command to call the unix utility ‘ps2pdf’ to convert the Postscript (.ps) file to a PDF (.pdf) file, the second name ‘outfilePDF’ is the expected output filename from the ‘ps2pdf’ utility when it is fed ‘outfile’ as its input file. It’s really only defined for messaging purposes so that we know what file to look for as output from the program.

If you want these filenames to represent the two versions of matchup data that are being compared, then the filename ‘outfile’ will have to be edited accordingly, or new parameters will need to be added to the calling sequence to allow the two versions or the Postscript file name to be specified as inputs. The following figures show examples of each page of the PDF output from the **agu\_nexrad\_bias\_maps\_dpr\_v3\_v4** program. Only the first 3 pages of output are shown, as page 4 is a repeat of page 1 to allow the viewer to toggle between any two graphics.

The following unix command launches a utility to allow you to view the PDF file:

**evince /data/tmp/map\_ITE109\_V4\_DPR\_bias.pdf &**

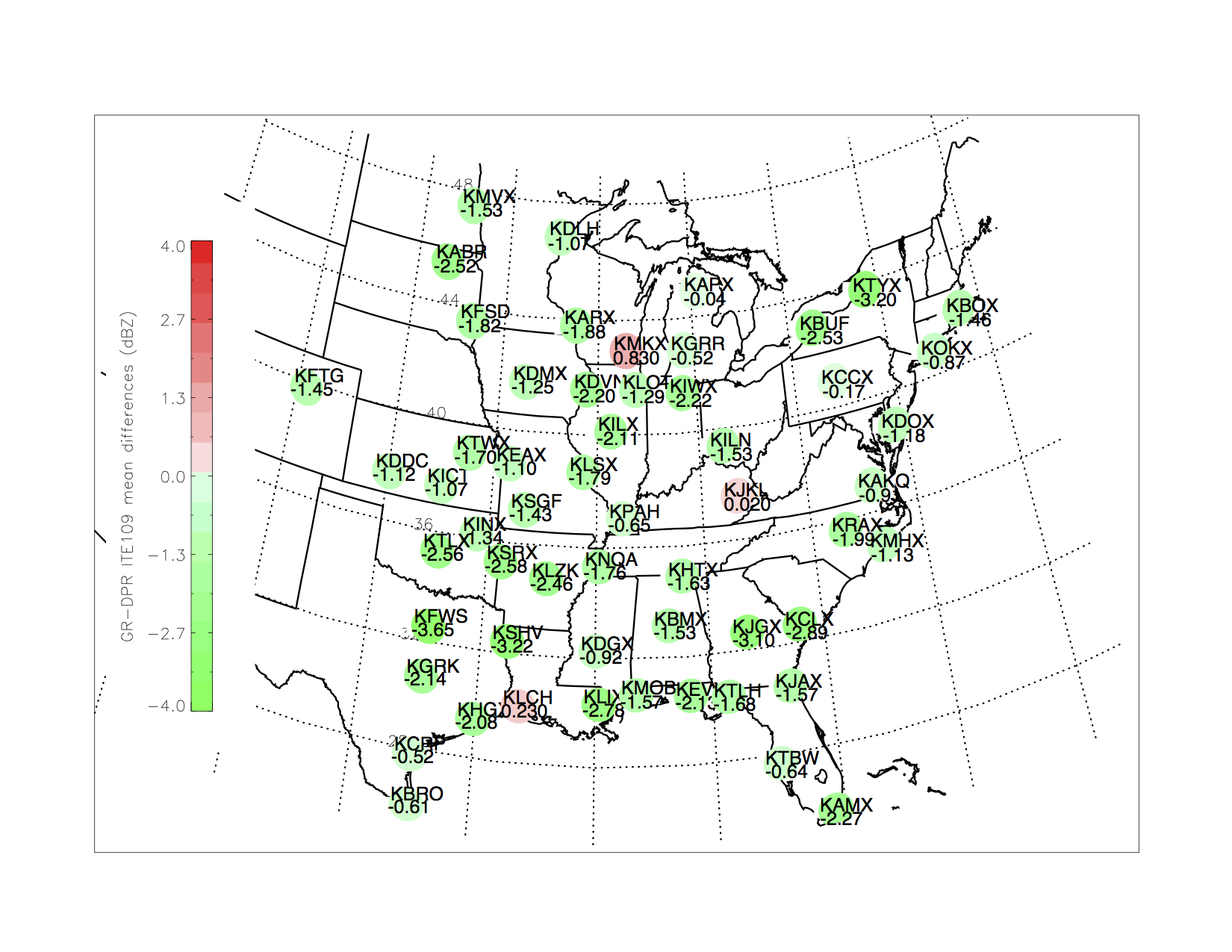


Figure . First page of output from agu\_nexrad\_bias\_maps\_dpr\_v3\_v4, showing GR-DPR mean Z bias by site for ITE109.

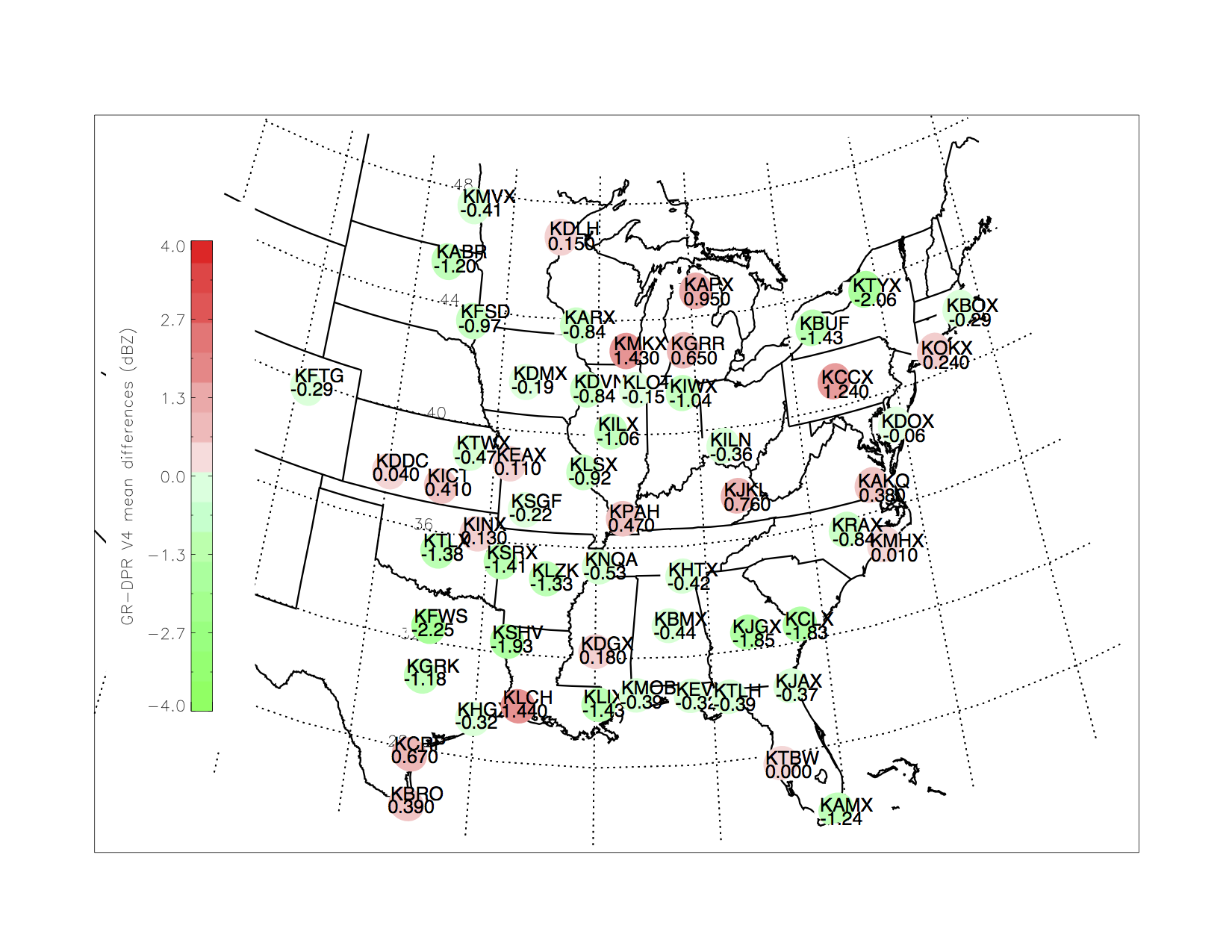


Figure . Second page of output from agu\_nexrad\_bias\_maps\_dpr\_v3\_v4, showing GR-DPR mean Z bias by site for V04A.

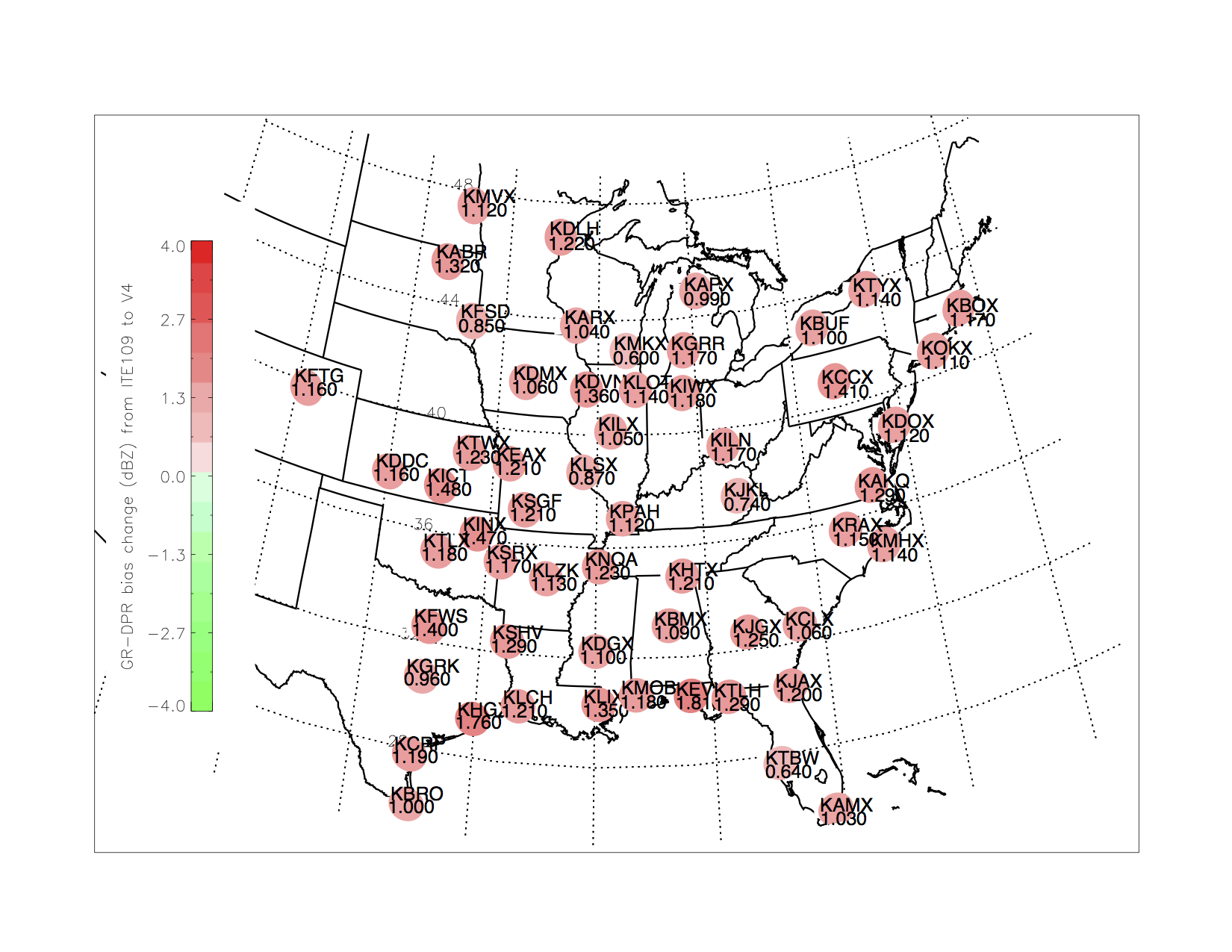


Figure . Third page of output from agu\_nexrad\_bias\_maps\_dpr\_v3\_v4, showing change in GR-DPR mean Z bias by site going from ITE109 to V04A. Positive values in this case indicate the increase in DPR Z values for ITE109.

## Producing Input Data for Time Series of Bias Plots

The IDL procedure **plot\_event\_series\_stacked\_color\_n\_sd** takes a file output by an SQL query operating on the tables loaded as described in Section 2.1. Assuming that the database table **zdiff\_stats\_dpr\_v4\_s2ku** has been loaded with data from the run of the IDL procedure **stats\_by\_dist\_to\_dbfile\_dpr\_pr\_geo\_match**, then start a psql session and run the following query to produce the data file for input to **plot\_event\_series\_stacked\_color\_n\_sd:**

=> psql gpmgv

psql (8.4.20)

Type "help" for help.

gpmgv=>

At the gpmgv=> prompt, enter and run the following SQL command to create a time series data file named /data/tmp/stats\_for\_db/event\_best\_diffsV4Pct100s2ku.txt:

\t \a \f '|' \o /data/tmp/stats\_for\_db/event\_best\_diffsV4Pct100s2ku.txt \\select a.radar\_id, date\_trunc('day',b.overpass\_time at time zone 'UTC') as event\_date, round((sum(meandiff\*numpts)/sum(numpts))\*100)/100 as zbias, sum(numpts) as total, round((sum(diffstddev\*numpts)/sum(numpts))\*100)/100 as bias\_stddev from zdiff\_stats\_dpr\_v4\_s2ku a, overpass\_event b where regime='S\_above' and a.orbit=b.orbit and a.radar\_id=b.radar\_id and b.sat\_id='GPM' and percent\_of\_bins=100 group by 1,2 order by 1,2;

Then enter \q to end the psql session, and start an IDL session to plot the time series data:

IDL> plot\_event\_series\_stacked\_color\_n\_sd, $

'/data/tmp/stats\_for\_db/event\_best\_diffsV4Pct100s2ku.txt', /READ\_STATIONS

This produces the following Postscript file:

/data/tmp/stats\_for\_db/event\_best\_diffsV4Pct100s2ku\_unsmoothed.ps

It is important to set the READ\_STATIONS binary parameter to On so that the code uses the station IDs included in the file in the time series plots. Otherwise it will ignore any radar ID not defined in a fixed list in the code. By default, the code will produce a Postscript file of time series plots, several time series per page, having the same file pathname as the input data file, but with the ‘.txt’ extension replaced with ‘.ps’, with an indicator of the state of the NSMOOTH option, and if set, of the GR\_MINUS\_PR option. For instance, if the above is re-run as so:

IDL> plot\_event\_series\_stacked\_color\_n\_sd, $

'/data/tmp/stats\_for\_db/event\_best\_diffsV4Pct100s2ku.txt', /READ\_STATIONS, $

nsmooth=3, /GR\_MINUS\_PR

then the code will indicate the following as its output Postscript file:

Finished, see PS file: /data/tmp/stats\_for\_db/event\_best\_diffsV4Pct100s2ku\_smoothed\_by\_3\_GR-PR.ps

The Postscript file is not converted to PDF by **plot\_event\_series\_stacked\_color\_n\_sd** so this can be done from the unix prompt:

cd /data/tmp/stats\_for\_db

ps2pdf event\_best\_diffsV4Pct100s2ku\_smoothed\_by\_3\_GR-PR.ps

after which the .pdf file can be viewed via the ***evince*** utility. The following is an example of one page of time series output.

