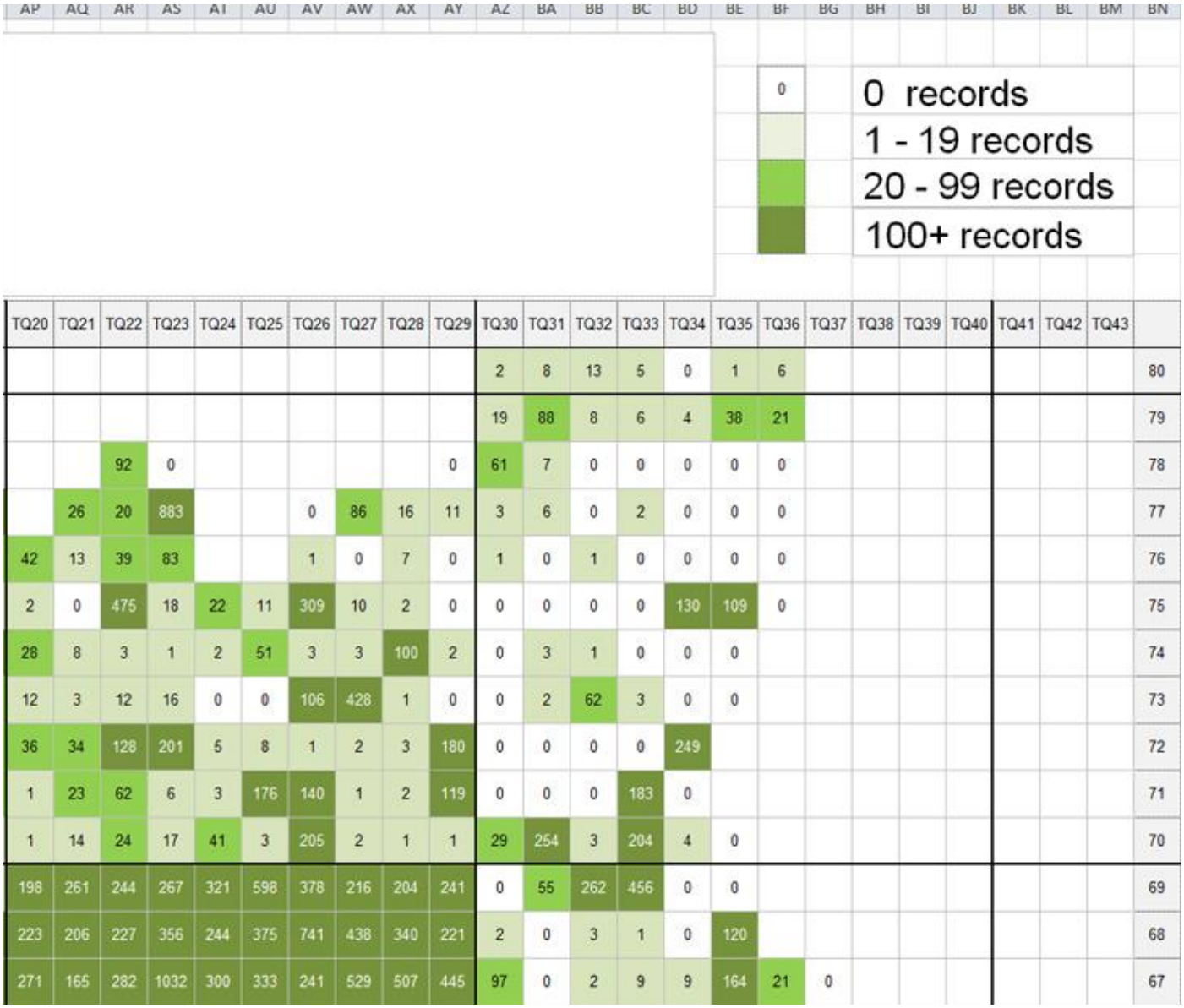
Appendix 2 - Records

MS Access 97 queries to generate a table of the number of records per 1km Square for a Vice County.

Note: To run these queries you will first need to download the 1kSquareList.xls file for the vice county of interest from http://www.ukflymines.co.uk/1KSquares/1kSquares_Co-ordinates.php and import it into your MapMate Records database.

Queries must be run in the order listed below. Copy and paste the green text into a separate new MS Access query MapMate Records database and save it with the name of the query in bold (include spaces if you type in the name or copy and paste the name).



TQ20	TQ21	TQ22	TQ23	TQ24	TQ25	TQ26	TQ27	TQ28	TQ29	TQ30	TQ31	TQ32	TQ33	TQ34	TQ35	TQ36	TQ37	TQ38	TQ39	TQ40	TQ41	TQ42	TQ43	
										2	8	13	5	0	1	6								80
										19	88	8	6	4	38	21								79
		92	0						0	61	7	0	0	0	0	0								78
	26	20	883			0	86	16	11	3	6	0	2	0	0	0								77
42	13	39	83			1	0	7	0	1	0	1	0	0	0	0								76
2	0	475	18	22	11	309	10	2	0	0	0	0	0	130	109	0								75
28	8	3	1	2	51	3	3	100	2	0	3	1	0	0	0									74
12	3	12	16	0	0	106	428	1	0	0	2	62	3	0	0									73
36	34	128	201	5	8	1	2	3	180	0	0	0	0	249										72
1	23	62	6	3	176	140	1	2	119	0	0	0	183	0										71
1	14	24	17	41	3	205	2	1	1	29	254	3	204	4	0									70
198	261	244	267	321	598	378	216	204	241	0	55	262	456	0	0									69
223	206	227	356	244	375	741	438	340	221	2	0	3	1	0	120									68
271	165	282	1032	300	333	241	529	507	445	97	0	2	9	9	164	21	0							67

Example: Detail of Analysis of the number of records of flowering plants in Surrey (VC 17) by 1km Square from 1666-2011.
Download an interactive version in MS Excel

Mapping Records - Step 1

```
SELECT Records.[_guk], Sites.[_guk], Taxa.[_guk], Taxa.Taxon, Sites.Name, Sites.OSGridRef,
IIf(Len([OSGridref])<=5,Null,IIf(Len([OSGridRef])=6,[OSGridRef],IIf(Len([OSGridRef])=8,Left([OSGridRef],4) &
Mid([OSGridRef],6,2),IIf(Len([OSGridRef])=10,Left([OSGridRef],4) &
Mid([OSGridRef],7,2),IIf(Len([OSGridRef])=12,Left([OSGridRef],4) &
Mid([OSGridRef],8,2),IIf(Len([OSGridRef])=14,Left([OSGridRef],4) & Mid([OSGridRef],9,2)))))) AS 1kSquare,
Int([_xo]/1000) AS X, Int([_yo]/1000) AS Y
```

FROM Taxa INNER JOIN (Records INNER JOIN Sites ON Records.[*Site] = Sites.[_guk]) ON Taxa.[_guk] = Records.[*Taxon]

GROUP BY Records.[_guk], Sites.[_guk], Taxa.[_guk], Taxa.Taxon, Sites.Name, Sites.OSGridRef, Int([_xo]/1000), Int([_yo]/1000), Records.Quantity

HAVING
(((IIf(Len([OSGridref])<=5,Null,IIf(Len([OSGridRef])=6,[OSGridRef],IIf(Len([OSGridRef])=8,Left([OSGridRef],4) & Mid([OSGridRef],6,2),IIf(Len([OSGridRef])=10,Left([OSGridRef],4) &
Mid([OSGridRef],7,2),IIf(Len([OSGridRef])=12,Left([OSGridRef],4) &
Mid([OSGridRef],8,2),IIf(Len([OSGridRef])=14,Left([OSGridRef],4) & Mid([OSGridRef],9,2)))))))) Is Not Null) AND ((Records.Quantity)<>-7))

ORDER BY Sites.OSGridRef;

Note: In MS Access the above queries the Site and Records tables to generate a Select Query of Records._guk, Sites._guk, OSGridRef, 1k Square and X and Y co-ordinates. Records where taxa are recorded as 'Not present' (Quantity = -7) are ignored. Once created you do not need to run this query as it is called by the next query.

Mapping Records - Step 2

SELECT DISTINCT Count(Records.[_guk]) AS CountOf_guk, [Mapping Records - Step 1].Sites.[_guk], [Mapping Records - Step 1].Taxa.[_guk], [Mapping Records - Step 1].[1kSquare], [Mapping Records - Step 1].X, [Mapping Records - Step 1].Y, Year([Date]) AS Year, Records.Date

FROM [Mapping Records - Step 1] INNER JOIN (Records INNER JOIN Taxa ON Records.[*Taxon] = Taxa.[_guk]) ON ([Mapping Records - Step 1].Records.[_guk] = Records.[_guk]) AND ([Mapping Records - Step 1].Sites.[_guk] = Records.[*Site])

GROUP BY [Mapping Records - Step 1].Sites.[_guk], [Mapping Records - Step 1].Taxa.[_guk], [Mapping Records - Step 1].[1kSquare], [Mapping Records - Step 1].X, [Mapping Records - Step 1].Y, Records.Date;

Note: In MS Access the above queries the Records, and Sites tables and the Mapping Records – Step 1 query to generate a Select Query of Records._guk, Sites._guk, Taxa._ukOSGridRef, 1k Square and X and Y co-ordinates, Year(Date) and Date. Once created you do not need to run this query as it is called by the next query.

Mapping Records - Step 3

SELECT [Mapping Records - Step 2].[1kSquare], [Mapping Records - Step 2].X, [Mapping Records - Step 2].Y, Sum([Mapping Records - Step 2].CountOf_guk) AS SumOfCountOf_guk INTO [Mapping Records - Table 1]

FROM [Mapping Records - Step 2]

WHERE ((([Mapping Records - Step 2].Year)>=[Start year] And ([Mapping Records - Step 2].Year)<=[End year]))

GROUP BY [Mapping Records - Step 2].[1kSquare], [Mapping Records - Step 2].X, [Mapping Records - Step 2].Y;

Note: In MS Access the above queries the Mapping Records – Step 2 query and prompts you to input the Start Year and End Year to generate a Select query which includes a list of 1k Squares, X and Y co-ordinates, Sum of Records for the Year Range (Start Year to End Year inclusive)

Mapping Records - Step 4

SELECT DISTINCT [1kSquareList].ID, [1kSquareList].[Vice-County], [Mapping Records - Table 1].[1kSquare], [Mapping Records - Table 1].X, [Mapping Records - Table 1].Y, [1kSquareList].ONE_KM, [1kSquareList].X, [1kSquareList].Y, [Mapping Records - Table 1].SumOfCountOf_guk INTO [Mapping Records - Table 2]

FROM 1kSquareList LEFT JOIN [Mapping Records - Table 1] ON [1kSquareList].ONE_KM = [Mapping Records - Table 1].[1kSquare]

WHERE ((([Mapping Records - Table 1].[1kSquare]) Is Null) AND (([Mapping Records - Table 1].X) Is Null) AND (([Mapping Records - Table 1].Y) Is Null));

Note: In MS Access the above queries the Mapping Records Table 1 – Table generated against the 1k SquareList table you downloaded and imported earlier to generate a new table of 1k Squares in your chosen Vice-County which do not have any records. This new table is Mapping Records – Table 2 – Table.

Mapping Records - Step 5

INSERT INTO [Mapping Records - Table 1] (1kSquare, X, Y)

SELECT [Mapping Records - Table 2].ONE_KM, [Mapping Records - Table 2].[1kSquareList_X], [Mapping Records - Table 2].[1kSquareList_Y]

FROM [Mapping Records - Table 2];

Note: In MS Access the above queries the 1k Squares without records (Totals = null) in Mapping Records Table 2 – Table are appended to the records Mapping Records Table 1 – Table.

Mapping Records - Step 6

UPDATE [Mapping Records - Table 1] SET [Mapping Records - Table 1].SumOfCountOf_guk = 0

WHERE ((([Mapping Records - Table 1].SumOfCountOf_guk) Is Null));

Note: In MS Access the above updates the null entries in Mapping Records Table 1 - Table **to zero**

Mapping Records - Step 7

SELECT [1kSquareList].ONE_KM, [1kSquareList].X, [1kSquareList].Y, [Mapping Records - Table 1].[1kSquare], [Mapping Records - Table 1].X, [Mapping Records - Table 1].Y, [Mapping Records - Table 1].SumOfCountOf_guk

FROM 1kSquareList RIGHT JOIN [Mapping Records - Table 1] ON [1kSquareList].ONE_KM = [Mapping Records - Table 1].[1kSquare]

WHERE ((([1kSquareList].ONE_KM) Is Null));

Note: In MS Access the above query generates a list of records in which the 1k Square lies outside the Vice-County. These are mostly probably due to an input error. You should review these records in 'Mapping Records Table 1 – Table' and either correct or delete them before proceeding to the final query.

Mapping Records - Step Final

TRANSFORM Sum([Mapping Records - Table 1].SumOfCountOf_guk) AS SumOfSumOfCountOf_guk

SELECT [Mapping Records - Table 1].Y

FROM [Mapping Records - Table 1]

GROUP BY [Mapping Records - Table 1].Y

ORDER BY [Mapping Records - Table 1].Y DESC

PIVOT [Mapping Records - Table 1].X;

Note: In MS Access the above query creates a cross-tab table, plotting the numbers of records in each 1km Square by eastings and northings (X and Y co-ordinates).

With the cursor in the top left-hand corner copy the table to the clipboard (Ctrl+C).

Open Excel and paste the table into a new blank worksheet (Ctrl-V) and save it.

You are now ready to customise the 'map' using Excel.