Supplementary materials for "Olfactory receptor subgenome and expression in a highly olfactory procellariiform seabird"

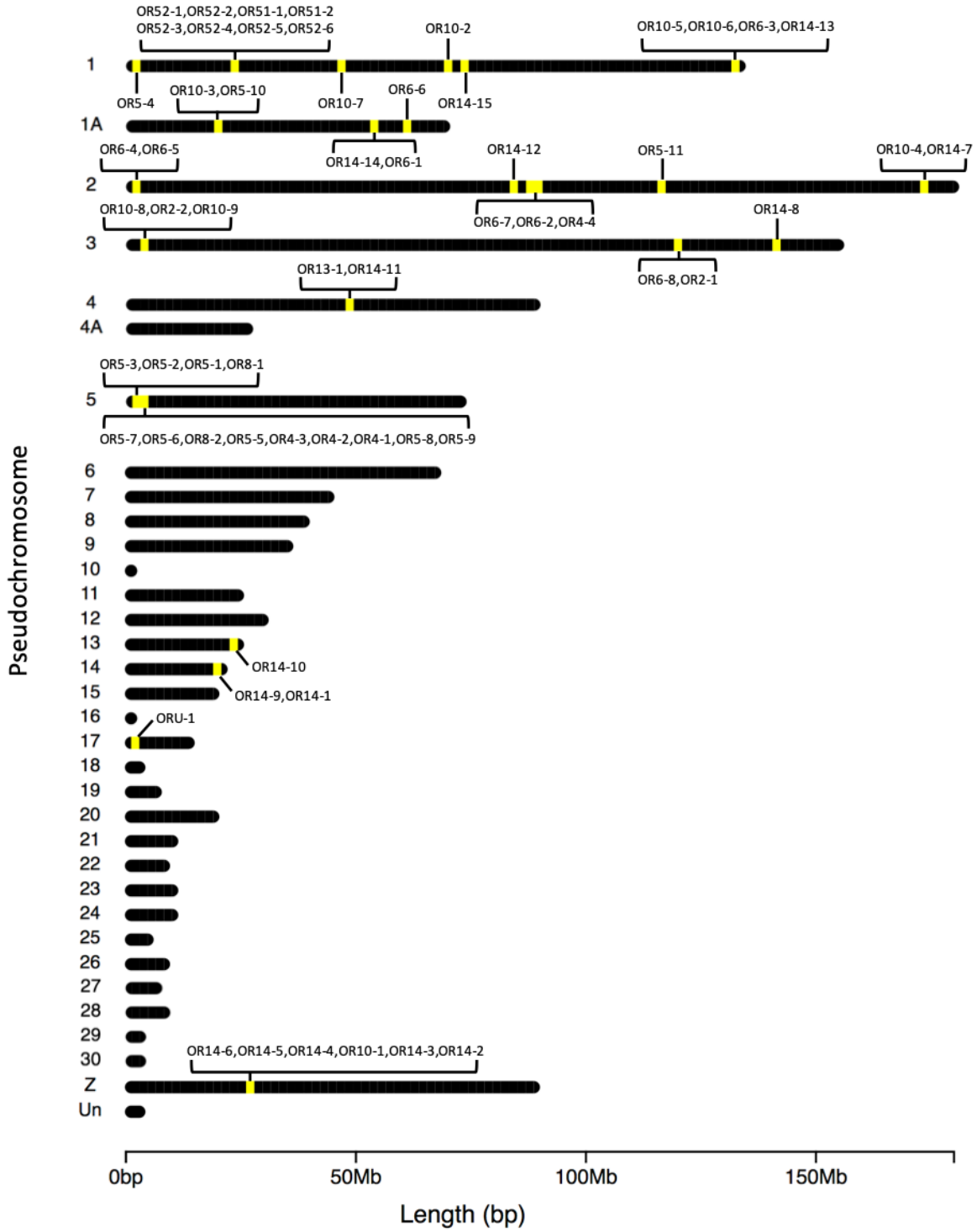

Figure S1. Genomic locations of OR genes on pseudochromosomes of the Leach's storm-petrel. Un indicates unplaced scaffolds.

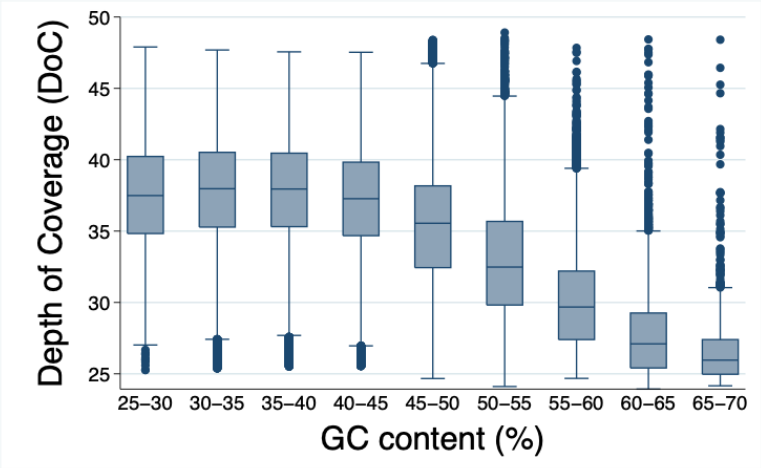

Figure S2. The Depth of Coverage (DoC) for bins of varying GC content.

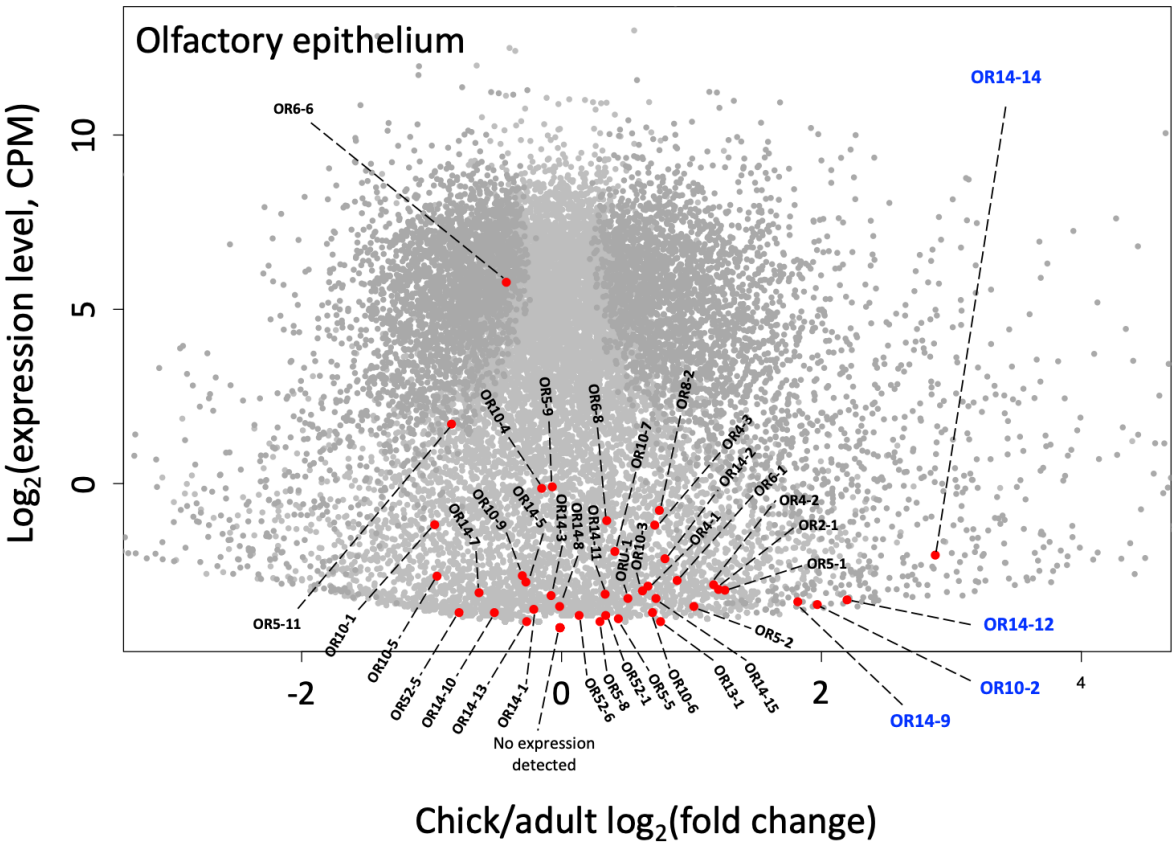

Figure S3. Differential expression of the genes in chick versus adult olfactory epithelium. Differentially expressed genes are in dark grey. OR genes are highlighted in red. Four OR genes with higher expression in chicks are labelled with their names in blue. Other OR genes are also labelled. Expression level is in counts per million (CPM).

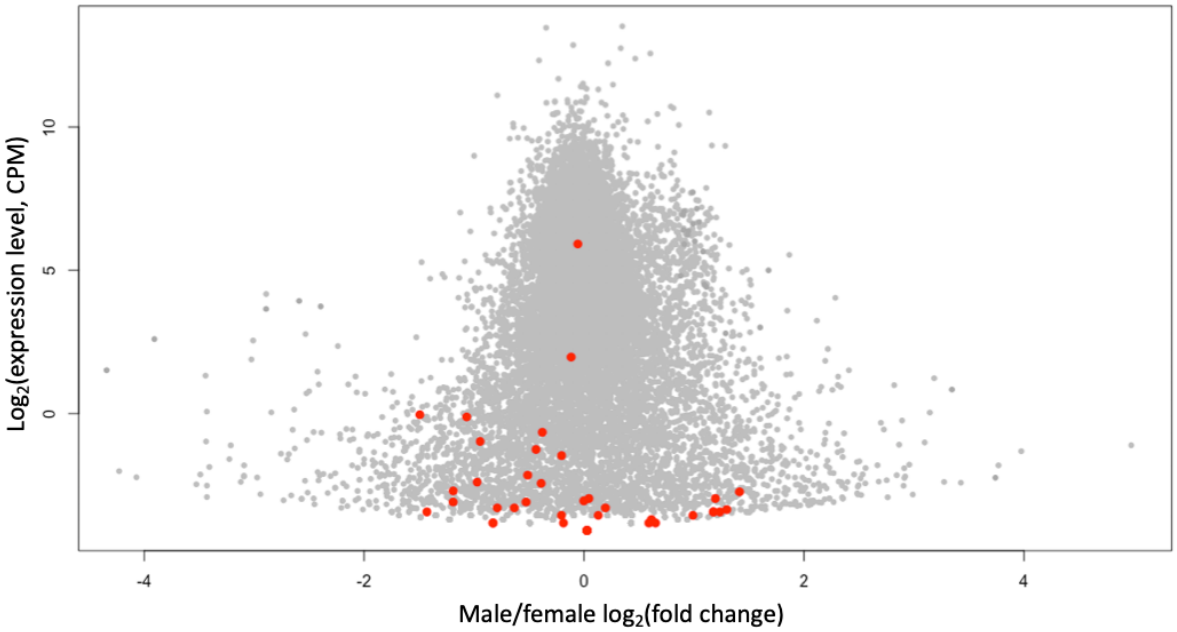

Figure S4. Differential expression of the genes in adult male versus female olfactory epithelium of the Leach's storm-petrel. Differentially expressed genes are in dark grey. OR genes are highlighted in red. No OR genes were differentially expressed between adult males and females. Expression level is in counts per million (CPM).

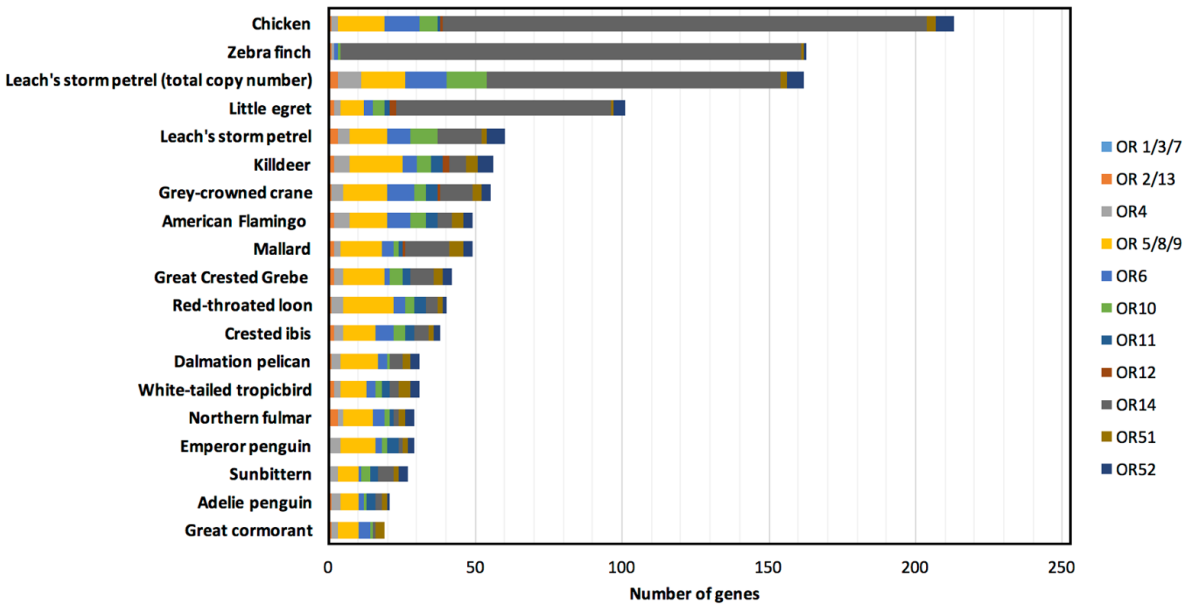

Figure S5. The number of intact OR genes partitioned by OR gene families in waterbirds, chicken, and zebra finch. Both the OR gene number estimations based on genome annotation and depth-of-coverage-assisted counting method (total copy number) in the Leach's storm-petrel are shown. The numbers for all species except the Leach's storm-petrel are from Khan *et al*. (*1*).

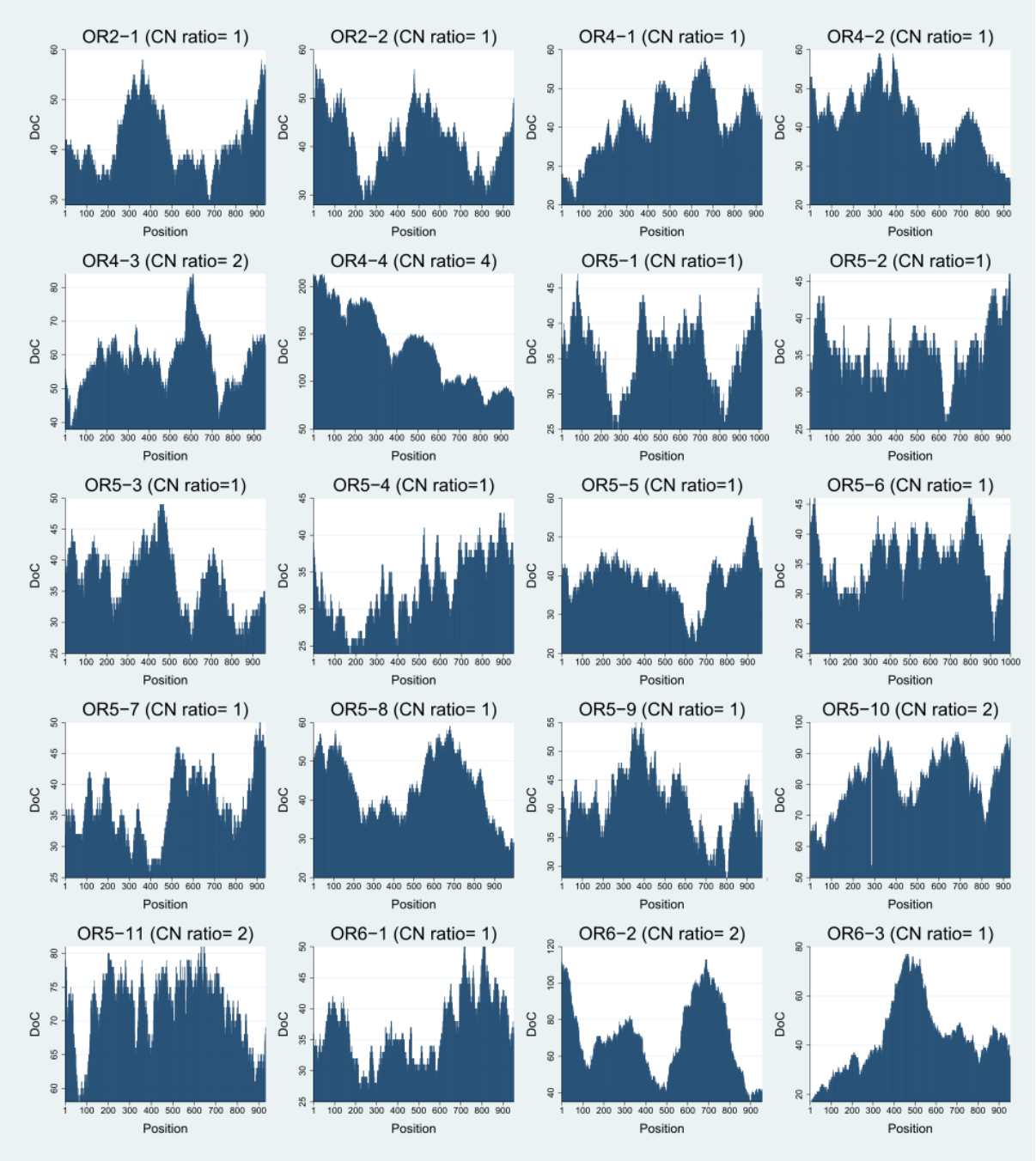

Figure S6 Depth of coverage (DoC) of mapped reads (y-axis) plotted against the sequence position (x-axis) for each intact olfactory receptor gene identified in the Leach's storm petrel. Sequence numbering begins at '1' (the first position in the coding region of intact OR genes), and storm petrel OR10-8 and OR10-9 (with few/no mapped reads) are omitted. Individual plots are drawn for each locus, with the locus name and copy number ratio (CN ratio) estimated from comparison of mean DoC to genome-wide averages for regions of similar GC-content indicated at the top of each plot.

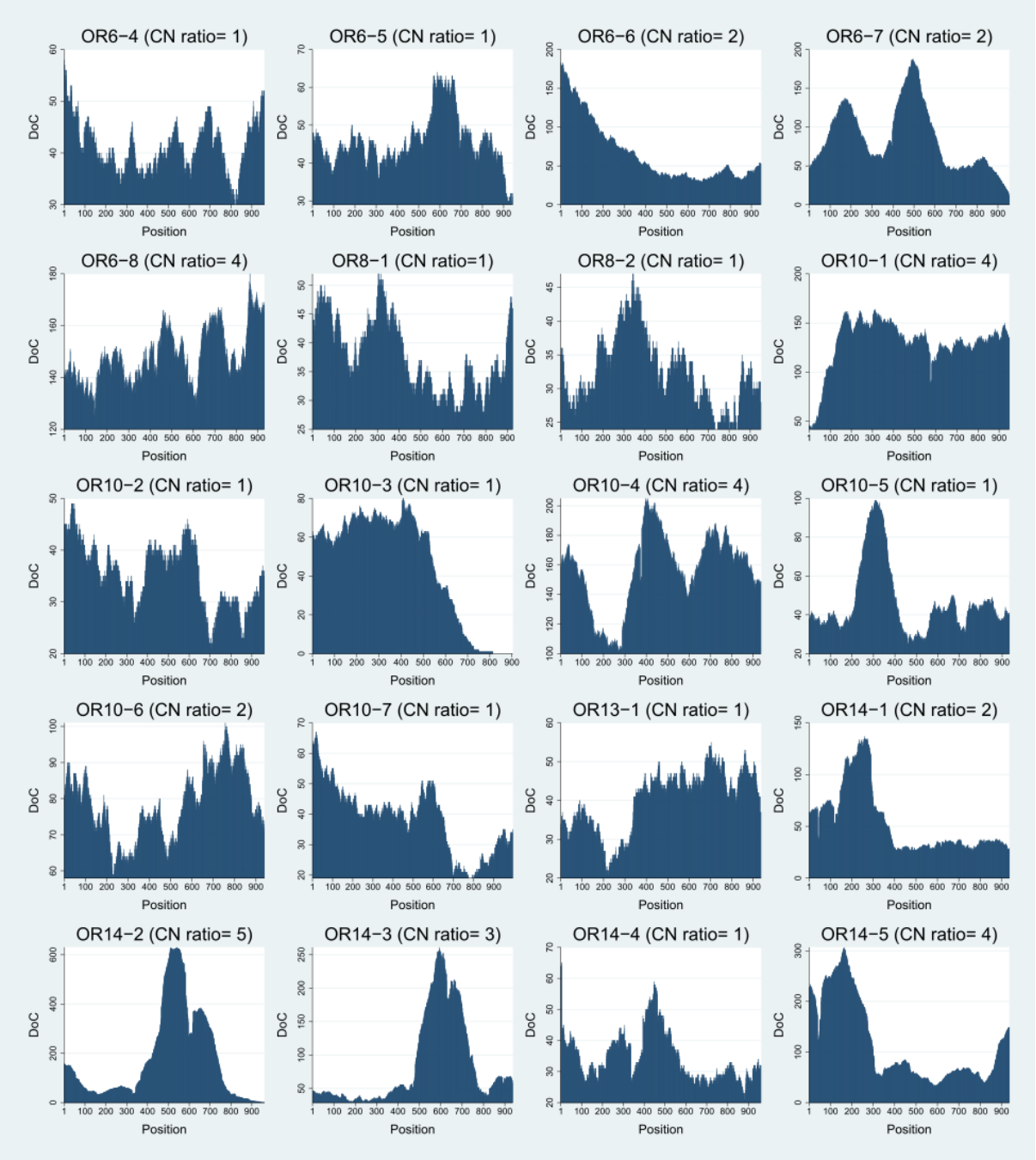

Figure S6 (continued)

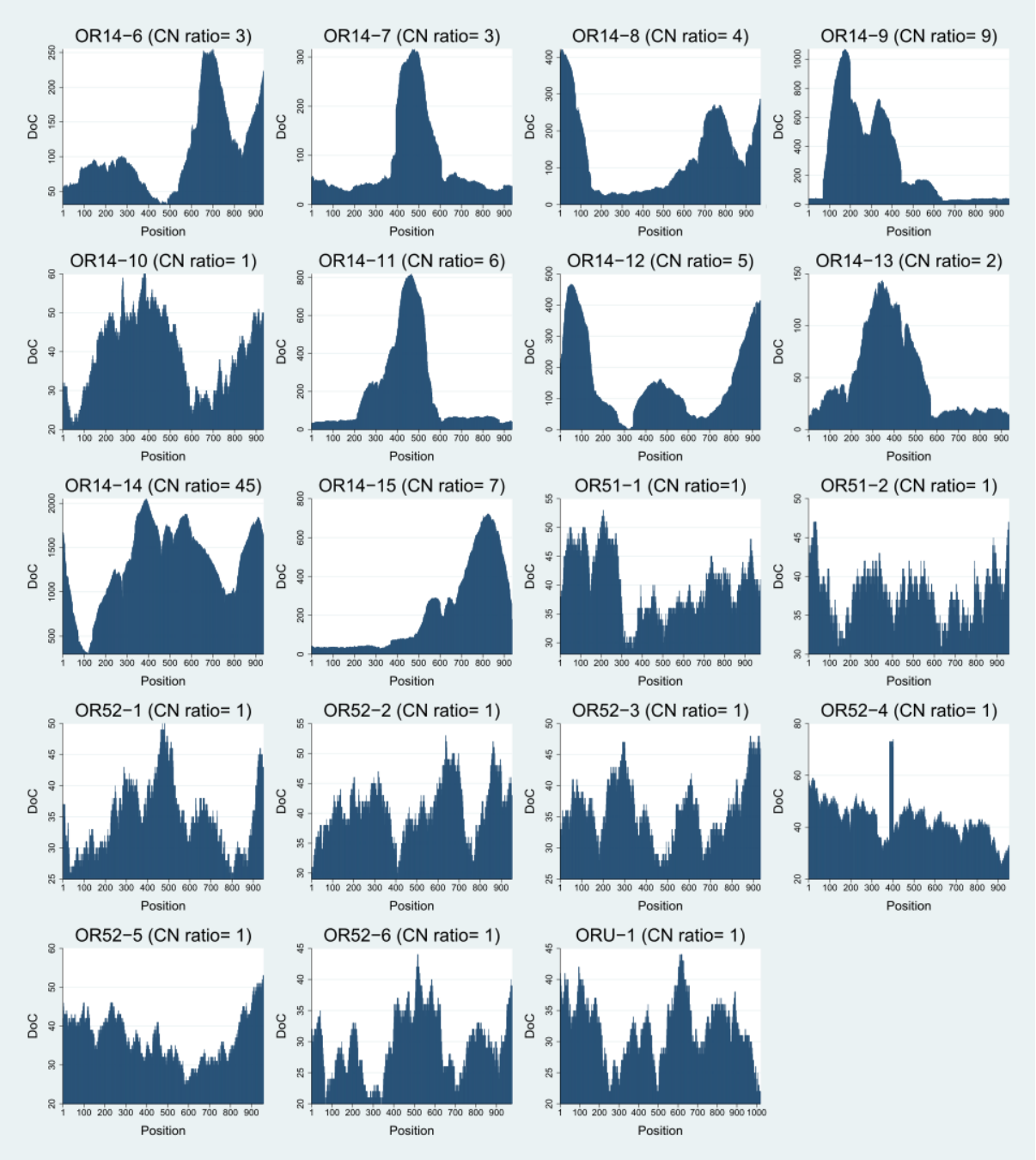

Figure S6 (continued)

Table S1 Information on tissues of Leach's storm-petrel sampled in this study.

MCZ[C] accession no.	Sampling date	Age class	Sex	Tissue type	Dataset[A]
Cryo149685	2 Sep 2015	Adult	F	Brain, olfactory concha	DE
Cryo149686	2 Sep 2015	Adult	M	Brain, olfactory concha, muscle, liver, heart, small intestine, kidney, tongue, eye, testes	DE, transciptome assembly, TopHat
Cryo149687	2 Sep 2015	Adult	M	Brain, olfactory concha	DE
				Stomach	TopHat
Cryo149688	3 Sep 2015	Chick	M? [B]	Brain, olfactory concha	DE
Cryo149689	3 Sep 2015	Adult	F	Brain, olfactory concha	DE
Cryo149690	3 Sep 2015	Adult	M	Brain, olfactory concha	DE
Cryo149691	4 Sep 2015	Chick	M? [B]	Brain, olfactory concha	DE
Cryo149692	4 Sep 2015	Adult	F	Brain, olfactory concha	DE
				Ovary	TopHat
Cryo149693	5 Sep 2015	Chick	M	Brain, olfactory concha	DE
				Spleen	TopHat

[A] DE: Differential expression analysis.
[B] Gonad too small to be certain.
[C] Museum of Comparative Zoology, Harvard University.

Table S2 Intact OR amino acid query sequences.

Species	Species name	No. of intact OR genes	Reference
Chicken	*Gallus gallus*	266	(*2*)
Zebra finch	*Taeniopygia guttata*	290	(*2*)
American alligator	*Alligator mississippiensis*	465	(*2*)
Saltwater crocodile	*Crocodylus porosus*	592	(*2*)
Indian gharial	*Gavialis gangeticus*	597	(*2*)
Burmese python	*Python bivittatus*	481	(*2*)
Green anole	*Anolis carolinensis*	108	(*2*)
Painted turtle	*Chrysemys picta*	842	(*2*)
Chinese softshell turtle	*Pelodiscus sinensis*	1180	(*2*)
Xenopus	*Xenopus tropicalis*	824	(*3*)
Zebrafish	*Danio rerio*	154	(*3*)
Human	*Homo sapiens*	388	HORDE database [A]

[A] HORDE database (build #44 2 Dec 2015, accessed Oct 24, 2018)

Table S3 Species included for phylogenetic analysis.

Species	Species name	Reference
American alligator	*Alligator mississippiensis*	(*2*)
Green anole	*Anolis carolinensis*	(*2*)
Chicken	*Gallus gallus*	(*2*)
Zebra finch	*Taeniopygia guttata*	(*2*)
Northern fulmar	*Fulmarus glacialis*	(*4*)
Emperor penguin	*Aptenodytes forsteri*	(*4*)
Adelie penguin	*Pygoscelis adeliae*	(*4*)
Great cormorant	*Phalacrocorax carbo*	(*4*)
Crested ibis	*Nipponia nippon*	(*4*)
Little egret	*Egretta garzetta*	(*4*)
Dalmatian pelican	*Pelecanus crispus*	(*4*)
Red-throated loon	*Gavia stellata*	(*4*)
White-tailed tropicbird	*Phaethon lepturus*	(*4*)
Sunbittern	*Eurypyga helias*	(*4*)
Killdeer	*Charadrius vociferus*	(*4*)
Grey-crowned crane	*Balearica regulorum*	(*4*)
Mallard	*Anas platyrhynchos*	(*4*)
Leach's storm-petrel	*Oceanodroma leucorhoa*	This study

Table S4 Assembly statistics for Leach's storm-petrel genome.

	Leach's storm-petrel genome
Estimated genome size	1.24 Gb
%GC content	42.1
Total depth of coverage	80x
Total contig length (bp)	1,181,786,487
Total scaffold length (bp, gapped)	1,195,165,757
Number of contigs	17396
Contig N50 (bp)	165.4 kb
Number of scaffolds	1697
Scaffold N50 (with gaps)	8.7 Mb
Total BUSCOs	4817/4915 (98.0%)
Complete BUSCOs	4654/4915 (94.7%)

Table S5 Copy number ratio of intact OR genes in the Leach's storm-petrel.

OR ID	OR family	Scaffold number	Scaffold length	Start position	End position	Copy number ratio	(+/- 2SD)[A]
OR2-1	OR2	373	42447	22343	23281	1	(0.9-1.4)
OR2-2	OR2	75	5450577	124020	124970	1	(1.0-1.7)
OR4-1	OR4	267	245379	38837	39763	1	(1.0-1.7)
OR4-2	OR4	267	245379	44800	45729	1	(0.9-1.4)
OR4-3	OR4	267	245379	49579	50532	2	(1.3-2.1)
OR4-4	OR4	300	133585	91777	92736	4	(3.1-4.7)
OR5-1	OR5	158	1660507	1639261	1640274	1	(0.9-1.5)
OR5-10	OR5	338	95658	51028	51960	2	(1.8-2.8)
OR5-11	OR5	378	41666	3678	4607	2	(1.7-2.7)
OR5-2	OR5	158	1660507	1644012	1644944	1	(0.9-1.5)
OR5-3	OR5	158	1660507	1655196	1656155	1	(0.9-1.5)
OR5-4	OR5	182	1123554	991888	992835	1	(1.0-1.5)
OR5-5	OR5	235	449237	423258	424226	1	(1.0-1.6)
OR5-6	OR5	235	449237	432355	433353	1	(0.9-1.5)
OR5-7	OR5	235	449237	438649	439587	1	(0.9-1.5)
OR5-8	OR5	267	245379	23268	24263	1	(1.0-1.6)
OR5-9	OR5	27	11256205	11240716	11241687	1	(0.9-1.5)
OR6-1	OR6	293	139987	31957	32907	1	(0.8-1.3)
OR6-2	OR6	300	133585	5445	6398	2	(1.6-2.4)
OR6-3	OR6	359	52130	38736	39689	1	(1.0-1.5)
OR6-4	OR6	409	29924	8476	9435	1	(0.9-1.4)
OR6-5	OR6	409	29924	24170	25111	1	(1.0-1.6)
OR6-6	OR6	480	21778	14483	15427	2	(1.5-2.3)
OR6-7	OR6	485	19036	201	1154	2	(1.9-2.9)
OR6-8	OR6	578	11286	3830	4759	4	(3.3-5.1)
OR8-1	OR8	158	1660507	1631284	1632207	1	(0.9-1.5)
OR8-2	OR8	235	449237	427586	428536	1	(0.9-1.4)
OR10-1	OR10	274	226073	128330	129277	4	(3.0-4.8)
OR10-2	OR10	297	126640	65519	66478	1	(0.8-1.2)
OR10-3	OR10	338	95658	76892	77794	1	(1.0-1.5)
OR10-4	OR10	345	61266	2418	3353	4	(3.6-5.8)
OR10-5	OR10	396	40047	4694	5626	1	(1.1-1.7)
OR10-6	OR10	463	21626	16656	17594	2	(1.9-3.2)
OR10-7	OR10	49	7629941	6884893	6885885	1	(0.9-1.6)
OR10-8	OR10	75	5450577	115075	116013	0[B]	(0.0-0.0)
OR10-9	OR10	75	5450577	130763	131701	0[B]	(0.0-0.1)
OR13-1	OR13	380	41223	13999	14934	1	(0.9-1.5)
OR14-1	OR14	263	246623	186713	187648	2	(1.3-2.2)
OR14-10	OR14	444	24419	12615	13547	1	(1.0-1.6)
OR14-11	OR14	482	19059	9009	9944	6	(4.9-8.2)
OR14-12	OR14	507	17189	13641	14576	5	(4.0-6.6)
OR14-13	OR14	513	16678	3768	4703	2	(1.2-2.1)
OR14-14	OR14	554	13022	9482	10417	45	(36.2-58.1)
OR14-15	OR14	581	11244	4964	5899	7	(5.4-9.1)
OR14-2	OR14	274	226073	53682	54641	5	(4.3-7.3)

OR14-3	OR14	274	226073	100514	101449	3	(2.1-3.5)
OR14-4	OR14	274	226073	149030	149989	1	(0.8-1.4)
OR14-5	OR14	274	226073	170063	170998	4	(2.9-4.8)
OR14-6	OR14	274	226073	191110	192045	3	(2.7-4.4)
OR14-7	OR14	325	87698	16076	17011	3	(2.1-3.6)
OR14-8	OR14	369	46204	20875	21843	4	(3.3-5.5)
OR14-9	OR14	384	39901	16443	17402	9	(7.2-12.0)
OR51-1	OR51	205	852617	362423	363397	1	(0.9-1.5)
OR51-2	OR51	205	852617	369024	369980	1	(1.0-1.6)
OR52-1	OR52	205	852617	313130	314077	1	(0.9-1.4)
OR52-2	OR52	205	852617	324416	325366	1	(1.1-1.8)
OR52-3	OR52	205	852617	376250	377182	1	(0.9-1.5)
OR52-4	OR52	205	852617	435628	436578	1	(1.1-1.8)
OR52-5	OR52	205	852617	501313	502269	1	(0.9-1.5)
OR52-6	OR52	205	852617	528440	529414	1	(0.8-1.3)
ORU-1	unassigned (OR1/2/10?)	50	7562026	6143206	6144225	1	(0.9-1.4)

[A] +/- 2 standard deviations (SD) of the copy number ratio.

[B] Genes that had low coverage close to 0 or no coverage with the 220 bp DNA library. They were conservatively assumed to have a copy number of 1 for all analyses.

Table S6 Gene ontology (GO) categories enriched among genes significantly differentially expressed (FDR < 0.01) in the olfactory epithelium between chick and adult Leach's storm-petrels. Only the top 40 GO terms are shown.

GO Term	Description	P-value	FDR q-value [A]	Enrichment [B]	N	B	n	b
GO:0030198	Extracellular matrix organization	3.03E-31	4.51E-27	12.93	12880	273	146	40
GO:0043062	Extracellular structure organization	1.67E-30	1.24E-26	11.86	12880	305	146	41
GO:0001503	Ossification	1.25E-16	6.22E-13	10.84	12880	99	288	24
GO:0030199	Collagen fibril organization	1.28E-16	4.78E-13	26.39	12880	40	183	15
GO:0032963	Collagen metabolic process	1.19E-13	3.54E-10	35.4	12880	46	87	11
GO:0032502	Developmental process	4.71E-11	1.17E-07	1.81	12880	3881	193	105
GO:0009887	Animal organ morphogenesis	1.22E-10	2.59E-07	5.68	12880	431	121	23
GO:0001958	Endochondral ossification	7.05E-10	1.31E-06	17.89	12880	25	288	10
GO:0036075	Replacement ossification	7.05E-10	1.17E-06	17.89	12880	25	288	10
GO:0030282	Bone mineralization	2.15E-09	3.20E-06	22.44	12880	42	123	9
GO:0048856	Anatomical structure development	6.15E-09	8.34E-06	1.96	12880	2660	193	78
GO:0031214	Biomineral tissue development	1.21E-08	1.50E-05	15.4	12880	68	123	10
GO:0007155	Cell adhesion	1.84E-08	2.11E-05	2.92	12880	687	250	39
GO:0022610	Biological adhesion	2.30E-08	2.45E-05	2.9	12880	693	250	39
GO:0030574	Collagen catabolic process	9.40E-08	9.34E-05	24.63	12880	30	122	7
GO:0032964	Collagen biosynthetic process	1.08E-07	1.01E-04	98.7	12880	6	87	4
GO:0009653	Anatomical structure morphogenesis	1.61E-07	1.41E-04	2.49	12880	1180	193	44
GO:0070208	Protein heterotrimerization	3.29E-07	2.72E-04	357.78	12880	12	9	3
GO:0002063	Chondrocyte development	4.77E-07	3.74E-04	26.83	12880	20	144	6
GO:0030500	Regulation of bone mineralization	6.42E-07	4.79E-04	14.94	12880	57	121	8
GO:0001501	Skeletal system development	8.15E-07	5.78E-04	13.23	12880	147	53	8
GO:0003433	Chondrocyte development involved in endochondral bone morphogenesis	8.28E-07	5.61E-04	172.5	12880	4	56	3
GO:0048513	Animal organ development	9.99E-07	6.48E-04	2.23	12880	1031	274	49
GO:0001649	Osteoblast differentiation	1.74E-06	1.08E-03	4.59	12880	92	519	17
GO:0071230	Cellular response to amino acid stimulus	2.49E-06	1.49E-03	12.51	12880	58	142	8
GO:0016043	Cellular component organization	2.65E-06	1.52E-03	1.2	12880	4168	1285	500
GO:0071840	Cellular component organization or biogenesis	2.88E-06	1.59E-03	1.2	12880	4209	1285	504
GO:0070167	Regulation of biomineral tissue development	3.52E-06	1.87E-03	12.17	12880	70	121	8
GO:0071229	Cellular response to acid chemical	4.10E-06	2.11E-03	5.48	12880	166	184	13
GO:0003414	Chondrocyte morphogenesis involved in endochondral bone morphogenesis	4.30E-06	2.14E-03	12.31	12880	17	431	7
GO:0003422	Growth plate cartilage morphogenesis	4.30E-06	2.07E-03	12.31	12880	17	431	7
GO:0003429	Growth plate cartilage chondrocyte morphogenesis	4.30E-06	2.00E-03	12.31	12880	17	431	7
GO:0090171	Chondrocyte morphogenesis	4.30E-06	1.94E-03	12.31	12880	17	431	7
GO:0009888	Tissue development	4.35E-06	1.91E-03	10.26	12880	478	21	8
GO:0043588	Skin development	6.50E-06	2.77E-03	18.14	12880	30	142	6
GO:0050793	Regulation of developmental process	7.45E-06	3.09E-03	1.9	12880	2033	193	58
GO:2000145	Regulation of cell motility	7.57E-06	3.05E-03	2.47	12880	718	254	35
GO:0060536	Cartilage morphogenesis	1.04E-05	4.09E-03	9.2	12880	26	431	8
GO:0035987	Endodermal cell differentiation	1.41E-05	5.38E-03	6.1	12880	25	844	10
GO:0050878	Regulation of body fluid levels	1.76E-05	6.55E-03	2.78	12880	287	419	26

[A] 'FDR q-value' is the correction of the above p-value for multiple testing using the Benjamini and Hochberg method (*5*). Namely, for the ith term (ranked according to p-value) the FDR q-value is (p-value * number of GO terms) / i.

[B] Enrichment (N, B, n, b) is defined as follows:
N - is the total number of genes
B - is the total number of genes associated with a specific GO term
n - is the number of genes in the top of the user's input list or in the target set when appropriate
b - is the number of genes in the intersection
Enrichment = (b/n) / (B/N)

Table S7 Positively selected sites detected by five approaches, along with integrated analysis, in genes of OR families 14, 5 and 6 in the Leach's storm-petrel. The sites detected by two or more methods are in bold and underlined.

OR family	No. of sequences	Positively selected sites				
		SLAC	FEL	MEME	FUBAR	Integrative
14	15	**4**, **107**	**4**, 38, 99, **107**, 110, 134, **156**, **200**, **250**	**4**, 6, 25, 47, 93, **107**, 154, **156**, 172, 182, 183, **200**, 203, 238, **250**, 254, 261, 306, 307, 311, 312	**107**	**4**, 6, 25, 38, 47, 93, 99, **107**, 110, 134, 154, **156**, 172, 182, 183, **200**, 203, 238, **250**, 254, 261, 306, 307, 311, 312
5	11	NA	**218**, 225, 307, 329	38, 93, 151, 180, 211, **218**, 241	NA	38, 93, 151, 180, 211, **218**, 225, 241, 307, 329
6	8	NA	**16**, **218**	**16**, 191, 202, **218**, 274, 289	156	**16**, 156, 191, 202, **218**, 274, 289

References:

1. I. Khan *et al.*, Olfactory receptor subgenomes linked with broad ecological adaptations in Sauropsida. *Mol. Biol. Evol.* **32**, 2832–2843 (2015).
2. M. Vandewege *et al.*, Contrasting patterns of evolutionary diversification in the olfactory repertoires of reptile and bird genomes. *Genome Biology and Evolution* **8**, 470–480 (2016).
3. Y. Niimura, On the origin and evolution of vertebrate olfactory receptor genes: comparative genome analysis among 23 chordate species. *Genome Biology and Evolution* **1**, 34–44 (2009).
4. E. Jarvis *et al.*, Whole-genome analyses resolve early branches in the tree of life of modern birds. *Science* **346**, 1320–1331 (2014).
5. Y. Benjamini, Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* **57**, 289–300 (1995).

Appendix 1

Scripts for bioinformatics analyses.

README

```
README
DEPTH OF COVERAGE (DoC) ANALYSIS OF OLFACTORY RECEPTOR (OR) GENES IN LEACH'S STORM-PETREL
==========================================================================================

Description: Calculate the depth of coverage (DoC) for each olfactory receptor (OR)
gene identified in the petrel genome assembly and compare to genome-wide DoC to
determine if any identified ORs represent collapsed gene copies, allowing an estimate
of the total expected number of petrel ORs.  DoC comparisons are made against a
repeatmasked version of the storm-petrel genome assembly, comparing the DoC of each
petrel OR gene against the mean DoC for similarly sized (1000 bp) genomic bins with
similar GC-content to that petrel OR gene.

Note that scripts were written specifically to calculate olfactory gene DoC in the storm-petrel,
and might contain hard-coded paths to input/output files and directories, as well as DoC values
calculated specifically for the storm-petrel.

------------------------------------------------------------------------------------
READ MAPPING & POST-PROCESSING OF MAPPED READS
------------------------------------------------------------------------------------
Only the 220bp fragment libraries were used for DoC analyses, but data from both the
rapid run & high output run were used.

1. Reads were trimmed with Trimmomatic v. 0.32 with options:
   ILLUMINACLIP:adapters.fa:2:30:10:1<:true

2. Trimmed reads were mapped with BWA v. 0.7.15 and default parameters, with separate jobs
   run for paired (1P/2P) and unpaired (1U/2U) reads.

   First, a BWA index was built for the (non-repeatmasked) storm-petrel genome assembly:

   bwa index LESP.fasta

   Then, reads were mapped to the genome assembly (with read group specified) e.g.:

   bwa mem /path/to/LESP_genome/bwa_index/LESP.fasta -t 8 \
   -R '@RG\tID:C57WP.5\tPL:ILLUMINA\tPU:C57WPANXXCGATGT.5\tSM:LESP\tLB:LESP_CGATGT_220bp'  \
   LESP_CGATGT_highoutput_220bp_trimmed_1P.fastq.gz LESP_CGATGT_highoutput_220bp_trimmed_2P.fastq.gz  \
   > LESP_highoutput_PE.sam

3. SAMtools v. 1.5 was used to post-process mapped reads and merge output BWA SAM files.
   Example workflow:

   a) Convert each BWA output SAM -> BAM, omitting unmapped reads and setting a minimim
      mapping quality (MAPQ= 30):

      samtools view -bh -F 4 -q 30 file1.bam file1.sam
      samtools view -bh -F 4 -q 30 file2.bam file2.sam ...etc. for each sam file

   b) Coordinate sort each BAM file from previous step:

      samtools sort -O bam -o file1.sorted.bam file1.bam
      samtools sort -O bam -o file2.sorted.bam file2.bam ...etc. for each bam file from step above

   c) Merge all BAM files:

      samtools merge LESP_merged.bam file1.sorted.bam file2.sorted.bam ...etc. (list all sorted bam files)

   d) Resort the merged BAM file:

      samtools sort -O bam -o LESP_merged.sorted.bam LESP_merged.bam
```

4. Duplicates were marked and removed with Picard v. 2.18.9

```
java -Xmx16g -jar /path/to/picard_2.18.9/picard.jar MarkDuplicates I=LESP_merged.sorted.bam  \
O=LESP_nodupes.bam M=LESP_marked_dup_metrics.txt REMOVE_DUPLICATES=true
```

CALCULATING MEAN DoC OF OLFACTORY RECEPTOR GENES

---

1. Per-base depth of coverage was output from the de-duplicated mapped reads with BEDTools v. 2.26.0 genomecov option:

```
bedtools genomecov -d -ibam LESP_nodupes.bam -g /path/to/LESP_genome/LESP.fasta  \
> LESP_bedtools_genomecov
```

2. The BEDTools 'genomecov' output file was split into per-scaffold files with a custom Perl script:

   Script: write_scaffold_DoC.pl

   Takes as input the 'LESP_bedtools_genomecov' file output in Step 1 above, and outputs a new directory called 'scaffold_DoC/' that contains one tab-delimited text file with DoC values for each scaffold

3. RepeatMasker v. 4.0.5 with RepeatMasker Library 'Complete Database 20160829' and query species 'vertebrata metazoa' was used to RepeatMask the storm petrel reference genome assembly:

```
RepeatMasker -pa 8 -x -species 'vertebrata metazoa' -dir RepeatMasker/ LESP.fasta
```

4. A custom Perl script was used to split non-repetitive regions of the repeatmasked reference storm-petrel genome assembly into 'bins' of 1000 bp and to calculate the GC-content (%GC) of each of these bins.

   Script: write_DoC_bins.pl

   Takes as input the repeatmasked version of the petrel genome assembly output from Step 3 above ('LESP.fasta.masked'), and outputs a BED-format file ('LESP_DoC_bins_raw.bed') with an entry for each genomic bin. Only genomic bins with >= 980 'called' ACGT bases are considered, and the ID field for each entry contains an indexed bin number followed by the %GC for that bin.

5. BEDTools v. 2.26.0 was used to omit any bins that overlap with predicted petrel OR genes (specified in a BED-format file 'LESP_OR_genes.bed'):

```
subtractBed -A -a LESP_DoC_bins_raw.bed -b LESP_OR_genes.bed \
> LESP_DoC_bins_final.bed
```

6. A custom Perl script was used to calculate the mean DoC within each genomic bin and to place bins in categories of 5% GC content (e.g. 0-5%, 5-10%, 10-15%, etc.)

   Script: calculate_genomic_bin_DoC.pl

   Takes as input the 'LESP_DoC_bins_final.bed' BED-format file output in Step 5 above, and the per-scaffold DoC text files output to scaffold_DoC/ directory output in Step 2 above. Outputs a tab-delimited text file 'LESP_bins_summary.txt' that lists the genomic coordinates, %GC, GC-category, and average DoC for each bin.

7. A custom Perl script was used to compare the mean DoC for each storm-petrel OR gene to the mean DoC of genomic bins with similar GC-content to that OR gene (i.e. DoC for an OR gene with 50% GC would be compared to the mean DoC of all genomic bins with 50% GC).

   Script: calculate_petrel_OR_DoC.pl

   Takes as input: the (non-repeatmasked) petrel genome assembly, a BED-format file listing the genomic coordinates of each petrel OR gene ('LESP_OR_genes.bed'), and the text files listing per-base DoC for each scaffold output to scaffold_DoC directory in Step 2 above. Outputs a tab-delimited text file ('LESP_OR_DoC_summary.txt') that lists summary information for each petrel OR gene, including the average DoC across the gene and a comparison of that DoC value to the mean value for genomic bins with the same GC-content (this ratio is expressed both as a 'raw' value, and also rounded to the nearest integer value).

write_scaffold_DoC.pl

```
#!/usr/bin/perl
use strict;
use warnings;

my $infile= 'LESP_bedtools_genomecov';
open(IN, "<$infile") || die "Cannot open $infile for reading: $!\n";

my $outdir= 'scaffold_DoC/';
unless (-e $outdir){
	mkdir $outdir || die "Cannot create $outdir: $!\n";
}

my $count= 0;
my $stored= '';
while(<IN>){
	chomp($_);
	my @split= split(/\t/, $_);
	my $scaffold= $split[0];
	if ($stored eq $scaffold){
		print OUT "$_\n";
		$count++;
		$stored= $scaffold;
	}
	else{
		close(OUT);
		my $outfile= $outdir . $scaffold;
		open(OUT, ">$outfile") || die "Cannot open $outfile for writing: $!\n";
		print OUT "$_\n";
		$count++;
		$stored= $scaffold;
	}
	if ($count=~ /^(\d+)(0{5})$/){
		print "Processed: $count lines\n";
	}
}
close(IN);
print "\nFinished writing: $count lines\n\n";
```

write_DoC_bins.pl

```
#!/usr/bin/perl
use strict;
use warnings;

use Bio::SeqIO;
use Bio::Seq;

#output file to hold results
my $outfile= 'LESP_DoC_bins_raw.bed';
open(OUT, ">$outfile") || die "Cannot open $outfile for writing: $!\n";

#input file of Repeatmasked genome
my $fasta= 'LESP.fasta.masked';
my $seqio_in= Bio::SeqIO-> new(-file=> $fasta, -format=> 'fasta');

my $count= 0;
my $bins= 0;
my @vals;
while (my $seqobj= $seqio_in-> next_seq()){
	$count++;
	my $id= $seqobj-> display_id();
	print "\nProcessing: $id (count: $count)\n";
	my $len= $seqobj-> length();
	my $stop= $len - 1000;

	#go through scaffold, grabbing 1000 bp at a time
	for (my $i= 1; $i<= $stop; $i= $i + 1000){
		#grab subseq
		my $i2= $i + 999;
		my $subseq= $seqobj-> subseq($i,$i2);
		$subseq= uc($subseq);
		#only keep ACGT
		$subseq=~ s/([^ACGT])//g;
		#only proceed if we have min. 980 ACGT characters in subseq
		my $cklen= length($subseq);
		if ($cklen >= 980){
			#print out this 'bin' to use in DoC calcs.
			#output is bed format, so make start 0-based
			my $start= $i - 1;
			$bins++;

			#calculate %GC for this bin
			my $gc= $subseq;
			$gc=~ s/([^GC])//g;
			my $gclen= length($gc);
			my $percgc= ($gclen/$cklen)*100;
			$percgc= sprintf("%.1f", $percgc);
			push(@vals, $percgc);

			print OUT "$id\t$start\t$i2\tbin$bins|$percgc\t.\t+\n";
		}
	}
	print "Done! Total bins= $bins\n";
}
close(OUT);
print "\nFinished processing: $count scaffolds\n";
print "Output: $bins bins for DoC calcs.\n";
my @sortedgc= sort {$a <=> $b} (@vals);
my $min= shift(@sortedgc);
my $max= pop(@sortedgc);
print "Min %GC: $min, Max %GC: $max\n\n";
```

calculate_genomic_bin_DoC.pl

```
#!/usr/bin/perl
use strict;
use warnings;

use Statistics::Basic qw(:all);

#store info. on DoC 'bins' of 1000bp
my $infile= 'LESP_DoC_bins_final.bed';
open(IN, "<$infile") || die "Cannot open $infile for reading: $!\n";

my %bins;
my $count= 0;
print "\nPreparing to process 1 Kb genome 'bins'...\n\n";
while(<IN>){
	chomp($_);
	my @split1= split(/\t/, $_);
	my ($scaff1, $start1, $end1, $id1)= ($split1[0], $split1[1], $split1[2], $split1[3]);
	#increment start to be 1-based
	$start1++;
	#parse id
	my @split2= split(/\|/, $id1);
	my ($bin1, $gc1)= ($split2[0], $split2[1]);
	#sort %GC into increments of 5% (e.g. 0-5%, 5-10%, etc.)
	my $category;
	for (my $i= 0; $i<= 100; $i= $i + 5){
		my $j= $i+5;
		if (($gc1 >= $i) && ($gc1 < $j)){
			$category= "$i-$j\%"
		}
	}
	#store in %bins, keyed by scaffold, then id, with
	#start, end, %GC, and category as values
	$bins{$scaff1}{$bin1}{'start'}= $start1;
	$bins{$scaff1}{$bin1}{'end'}= $end1;
	$bins{$scaff1}{$bin1}{'gc'}= $gc1;
	$bins{$scaff1}{$bin1}{'category'}= $category;

	$count++;
	if ($count=~ /^(\d+)(0{4})$/){
		print "Processed: $count\n";
	}
}
close(IN);
print "\nFinished parsing: $count 'bins'\n";

open(OUT, ">LESP_bins_summary.txt") || die "Cannot open outfile for writing: $!\n";
print OUT "ID\tScaffold\tStart\tEnd\t%GC\tCategory\tAvg. DoC\n";

#get the average depth of coverage for each 'bin'
print "Calculating average DoC for each 'bin'...\n";
my $count2= 0;
for my $skey (sort {$a cmp $b} (keys %bins)){
	print "Processing: $skey\n";
	#store all of the DoC values for this scaffold
	my %doc;
	my $infile2= "scaffold_DoC/$skey";
	open(IN2, "<$infile2") || die "Cannot open $infile2 for reading: $!\n";
	print "Storing DoC values...\n";
	while(<IN2>){
		chomp($_);
		my @split3= split(/\t/, $_);
		my ($pos3, $depth3)= ($split3[1], $split3[2]);
		$doc{$pos3}= $depth3;
	}
	close(IN2);
	print "Calculating average DoC for each 'bin' on $skey...\n";
	for my $bkey (sort {$a cmp $b} (keys %{$bins{$skey}})){
		my $startval= $bins{$skey}{$bkey}{'start'};
		my $endval= $bins{$skey}{$bkey}{'end'};
		my @vals;

		for (my $z= $startval; $z<= $endval; $z++){
```

```
                                        my $dval= $doc{$z};
                                        push(@vals, $dval);
                                }
                                my $avg= mean(\@vals);
                                #output info. for this 'bin'
                                print OUT
"$bkey\t$skey\t$startval\t$endval\t$bins{$skey}{$bkey}{'gc'}\t$bins{$skey}{$bkey}{'category'}\t$avg\n";

                                $count2++;
                        }
}
close(OUT);
print "\nFinished calculating average DoC for $count2 'bins' total\n\n";
```

calculate_petrel_OR_DoC.pl

```
#!/usr/bin/perl
use strict;
use warnings;

use Bio::DB::Fasta;
use Bio::Seq;
use Statistics::Basic qw(:all);
use Math::Round qw(:all);

#outfile to hold results
my $outfile= 'LESP_OR_DoC_summary.txt';
open(OUT, ">$outfile") || die "Cannot open $outfile for writing: $!\n";
print OUT
"ID\tScaffold\tStart\tEnd\tStrand\tLength\tACGT_length\t%GC\tGC_bin\tAvg_DoC\tGC_bin_avg_Doc\tDoC_Ratio\t \
DoC_Ratio_integer\n";

#create a Fastadb of the petrel genome assembly
my $fasta= "LESP.fasta";
print "\nCreating fastadb of $fasta...\n";
my $fastadb= Bio::DB::Fasta-> new($fasta);
print "Done!\n\n";

#store the mean DoC averaged across all bins within a GC-category
#note: these hard-coded values were specifically determined by averaging across GC-bins
#for the storm-petrel genome
my %genomedoc;
$genomedoc{'25-30%'}= 37.4;
$genomedoc{'30-35%'}= 37.8;
$genomedoc{'35-40%'}= 37.8;
$genomedoc{'40-45%'}= 37.1;
$genomedoc{'45-50%'}= 35.4;
$genomedoc{'50-55%'}= 32.7;
$genomedoc{'55-60%'}= 30.1;
$genomedoc{'60-65%'}= 27.7;
$genomedoc{'65-70%'}= 26.7;

#store info for each petrel OR identified in the genome assembly
my %petrel;
my $orbed= 'LESP_OR_genes.bed';
open(BED, "<$orbed") || die "Cannot open $orbed for reading: $!\n";

my $count1= 0;
print "Storing coordinate info. for petrel ORs...\n";
while(<BED>){
        chomp($_);
        my @split1= split(/\t/, $_);
        my ($scaff1, $start1, $end1, $id1, $strand1)= ($split1[0], $split1[1], $split1[2], $split1[3], $split1[5]);
        #make start 1-based
        $start1++;
        #store keyed by scaffold, then ID
        $petrel{$scaff1}{$id1}{'start'}= $start1;
        $petrel{$scaff1}{$id1}{'end'}= $end1;
        $petrel{$scaff1}{$id1}{'strand'}= $strand1;
        $count1++;
}
close(BED);
print "\nFinished storing info. for: $count1 petrel ORs\n\n";

print "Calculating %GC for each petrel OR...\n";
my @ckvals;
my $count2= 0;
for my $skey (sort {$a cmp $b} (keys %petrel)){
        #retrieve a SeqObject for this scaffold
         my $seqobj= $fastadb-> get_Seq_by_id($skey);
        #for each OR, retrieve the scaffold subsequence & reverse complement if necessary before calculating GC
        for my $ikey (sort {$a cmp $b} (keys %{$petrel{$skey}})){
                my $pos1= $petrel{$skey}{$ikey}{'start'};
                my $pos2= $petrel{$skey}{$ikey}{'end'};
                my $strand= $petrel{$skey}{$ikey}{'strand'};

                my $subseq= $seqobj-> subseq($pos1,$pos2);
                $subseq= uc($subseq);
```

```
                        my $totlen= length($subseq);
                        $petrel{$skey}{$ikey}{'length'}= $totlen;
                        if ($strand eq '-'){
                                my $tmpobj= Bio::Seq-> new(-id=> 'tmp', -seq=> $subseq, -alphabet=> 'dna');
                                $tmpobj= $tmpobj-> revcom();
                                $subseq= $tmpobj-> seq();
                        }

                        #replace any non-ACGT characters
                        $subseq=~ s/([^ACGT])//g;
                        my $acgtlen= length($subseq);
                        $petrel{$skey}{$ikey}{'acgt_length'}= $acgtlen;

                        #calculate %GC
                        $subseq=~ s/([^GC])//g;
                        my $gclen= length($subseq);
                        my $gc= ($gclen/$acgtlen)*100;
                        $gc= sprintf("%.1f", $gc);
                        $petrel{$skey}{$ikey}{'perc_gc'}= $gc;
                        push(@ckvals, $gc);

                        #and, assign this %GC to a 'bin'
                        my $category;
                        for (my $i= 0; $i<= 100; $i= $i+5){
                                my $j= $i+5;
                                if (($gc >= $i) && ($gc < $j)){
                                        $category= "$i-$j\%"
                                }
                        }
                        $petrel{$skey}{$ikey}{'gc_bin'}= $category;
                        $count2++;
                }
}
print "\nFinished calculating %GC for $count2 petrel ORs\n\n";

my @sorted= sort {$a <=> $b} @ckvals;
my $min= shift(@sorted);
my $max= pop(@sorted);
print "Min %GC= $min, Max %GC= $max\n\n";

#finally, calculate avg. DoC for each petrel OR
print "\nCalculating avg DoC for petrel ORs...\n";
my $total2= 0;
for my $skey2 (sort {$a cmp $b} (keys %petrel)){
        print "Processing ORs on: $skey2\n";
        #store all of the DoC values for this scaffold
        my %doc;
        my $infile= "scaffold_DoC/$skey2";
        open(IN2, "<$infile") || die "Cannot open $infile for reading: $!\n";
        print "Storing DoC values...\n";
        while(<IN2>){
                chomp($_);
                my @split3= split(/\t/, $_);
                my ($pos3, $depth3)= ($split3[1], $split3[2]);
                $doc{$pos3}= $depth3;
        }
        close(IN2);
        print "Calculating average DoC for each petrel OR on $skey2...\n";
        for my $ikey2 (sort {$a cmp $b} (keys %{$petrel{$skey2}})){
                my $startval= $petrel{$skey2}{$ikey2}{'start'};
                my $endval= $petrel{$skey2}{$ikey2}{'end'};
                my @vals;

                for (my $z= $startval; $z<= $endval; $z++){
                        my $dval= $doc{$z};
                        push(@vals, $dval);
                }
                my $avg= mean(\@vals);

                #get the avg DoC for this GC bin
                my $binval= $petrel{$skey2}{$ikey2}{'gc_bin'};
                my $bdoc= $genomedoc{$binval};

                my $ratio2= $avg/$bdoc;
                $ratio2= sprintf("%.1f", $ratio2);
```

```
                        my $int2= nearest_ceil(1, $ratio2);
                        $total2+=$int2;

                        print OUT
"$ikey2\t$skey2\t$startval\t$endval\t$petrel{$skey2}{$ikey2}{'strand'}\t$petrel{$skey2}{$ikey2}{'length'}\t \
$petrel{$skey2}{$ikey2}{'acgt_length'}\t$petrel{$skey2}{$ikey2}{'perc_gc'}\t$binval\t$avg\t$bdoc\t$ratio2\t$int2\n";
                }
}
close(OUT);
print "Estimated total OR gene content based on GC-bin average DoC= $total2\n\n";
```