Supplementary Material

TABLE S1. Summary of sequencing of bacterial 16S rRNA grouped according to sample type and time point.

			Prior to sequencing		Prior to normalization				After to normalization		
Sample[2]	**Time[3]**	**n**	**DNA (ng/µL)**	**Total DNA yield (µg)[1]**	**Sequences (Total)**	**Sequences (Mean±SD)**	**OTUs (Mean±SD)**	**Coverage (Mean±SD)**	**Sequences (Total)**	**Sequences (Mean±SD)**	**OTUs (Mean±SD)**
BS	T1	6	20.54±22.52	616.30±675.47	68991	11498.50±4400.99	1098.83±248.77	0.96±0.01	36410	6068.33±160.80	963.00±226.42
	T2	8	14.95±17.93	448.53±537.78	84719	10589.88±1607.08	655.88±322.80	0.98±0.01	48704	6088.00±104.73	637.00±343.45
	T3	8	10.18±12.72	305.45±381.61	84905	10613.12±2309.19	448.50±156.32	0.98±0.01	48270	6033.75±68.12	412.62±184.68
	T4	8	26.28±18.59	788.43±557.84	100017	12502.12±4226.33	1008.00±379.11	0.97±0.01	48071	6008.88±86.08	857.50±249.40
	T5	8	14.33±17.51	429.89±525.26	77074	9634.25±2081.29	699.25±400.44	0.97±0.01	48363	6045.38±94.89	651.75±348.62
	T6	4	13.43±18.90	402.99±567.00	29983	7495.75±358.15	809.50±363.50	0.96±0.01	24405	6101.25±46.56	809.50±363.50
RAL	T1	8	650.92±114.48	32546.02±5723.95	131075	16384.38±3754.56	1203.75±194.92	0.98±0.01	47857	5982.12±133.81	907.50±131.57
	T2	8	569.52±121.13	28475.77±6056.58	141771	17721.38±4530.15	1230.12±184.70	0.98±0.01	47058	5882.25±79.50	838.88±124.49
	T3	8	416.78±95.54	20838.80±4777.00	113912	14239.00±3509.42	1192.50±172.66	0.97±0.01	48570	6071.25±180.75	990.75±137.25
	T4	8	567.11±143.18	28355.59±7159.03	118150	14768.75±5114.35	1175.88±200.94	0.97±0.01	48397	6049.62±142.81	977.75±133.16
	T5	8	537.55±124.43	26877.74±6221.69	100765	12595.62±3257.00	1083.00±121.18	0.97±0.01	48051	6006.38±200.24	897.50±142.46
	T6	8	589.64±147.00	29482.12±7349.79	111456	13932.00±5472.65	1103.50±173.43	0.97±0.01	47481	5935.12±161.87	856.62±194.65
RAS	T1	8	651.30±46.83	32565.05±2341.27	285748	35718.50±72325.79	1088.38±496.28	0.97±0.01	47830	5978.75±144.40	832.75±112.90
	T2	8	626.01±117.91	31300.53±5895.27	100766	12595.75±2851.55	1037.25±169.40	0.97±0.01	47352	5919.00±107.47	803.75±149.53
	T3	8	552.58±77.28	27629.05±3863.92	93596	11699.50±3677.48	1011.38±173.94	0.97±0.01	48549	6068.62±107.41	930.00±116.91
	T4	8	538.69±75.12	26934.36±3756.05	104338	13042.25±5455.85	1038.38±216.27	0.97±0.01	48381	6047.62±85.87	930.12±178.06
	T5	8	512.80±45.91	25640.13±2295.43	83244	10405.50±3110.43	959.00±153.78	0.97±0.01	48770	6096.25±111.07	917.50±127.94
	T6	8	533.85±75.55	26692.69±3777.66	76529	9566.12±2854.20	907.75±141.64	0.97±0.01	48175	6021.88±146.47	830.00±103.19
RVL	T1	7	564.10±122.64	28205.00±6132.18	126684	18097.71±10043.96	1200.71±255.40	0.98±0.01	41315	5902.14±66.71	831.29±138.54
	T2	8	562.40±122.64	28120.15±6132.19	148868	18608.50±5020.85	1238.88±214.89	0.98±0.00	47423	5927.88±45.24	842.38±142.96
	T3	8	474.55±70.51	23727.53±3525.48	125867	15733.38±3153.03	1181.75±193.47	0.98±0.00	47476	5934.50±98.97	843.25±165.10
	T4	8	538.49±121.67	26924.55±6083.44	112302	14037.75±4402.51	1162.38±181.56	0.97±0.01	47166	5895.75±123.63	831.38±136.47
	T5	8	573.73±108.96	28686.67±5447.87	128424	16053.00±4386.48	1182.75±187.49	0.98±0.01	47217	5902.12±85.91	823.25±108.32
	T6	8	532.81±168.68	26640.70±8433.98	100150	12518.75±3909.30	1063.00±176.95	0.97±0.01	48269	6033.62±149.57	912.50±142.71
RVS	T1	8	638.58±73.60	31928.93±3679.78	113279	14159.88±4906.35	1046.75±243.62	0.98±0.01	47868	5983.50±127.61	838.00±164.77
	T2	8	666.14±46.39	33307.14±2319.44	107639	13454.88±3884.61	1073.88±213.81	0.98±0.01	47746	5968.25±147.24	867.88±150.49
	T3	8	603.31±111.16	30165.41±5557.79	95357	11919.62±1837.64	1046.38±169.67	0.97±0.00	47868	5983.50±205.84	879.88±155.16
	T4	8	553.95±74.95	27697.26±3747.46	76518	9564.75±2040.66	936.12±120.28	0.97±0.01	48379	6047.38±79.20	899.75±121.80
	T5	8	502.22±69.15	25111.05±3457.46	81654	10206.75±2446.30	956.88±113.38	0.97±0.01	48175	6021.88±131.83	879.62±134.55
	T6	7	547.89±40.61	27394.40±2030.35	60154	8593.43±956.39	900.57±130.62	0.97±0.01	42440	6062.86±52.70	900.57±130.62

[1]Total DNA yield was calculated by multiplying DNA concentration measured with the Qubit platform with final elution volume. Eluted volume for buccal swab and rumen samples was 30 µL and 50 µL, respectively. [2]BS: buccal swab, RAL: rumen anterior liquid, RAS: rumen anterior solid, RVL: rumen ventral liquid and RVS: rumen ventral solid; [3]Sampling was performed every 2 hours over the course of 10 hours, starting 1 hour prior to morning feeding (~ 9 AM) and ending just prior to evening feeding (~ 7 PM), totaling six time points (T1-T6).

TABLE S2. Summary of analysis of variance of bacterial alpha diversity.

Alpha index	Factor	numDF	denDF	F value	P-value
Sobs[1]	(Intercept)	1	160	43568.71	0.000
	Time[4]	5	35	2.63	0.040
	Class[5]	4	160	15.30	0.000
	Time:Class[6]	20	160	3.82	0.000
Shannon[2]	(Intercept)	1	160	1789.25	0.000
	Time	5	35	2.27	0.069
	Class	4	160	29.72	0.000
	Time:Class	20	160	3.84	0.000
Invsimpson[3]	(Intercept)	1	160	3517.45	0.000
	Time	5	35	2.56	0.044
	Class	4	160	12.30	0.000
	Time:Class	20	160	4.95	0.000

Index	Sampling Time	Mean	SE	lower.CL	upper.CL	Tukey HSD
Sobs[1]	T1	871.27	46.38	718.56	1056.43	a
	T2	789.23	41.01	653.89	952.57	a
	T3	776.58	40.36	643.42	937.31	a
	T4	894.54	46.49	741.15	1079.69	a
	T5	824.38	42.84	683.02	995.00	a
	T6	856.59	46.80	702.88	1043.91	a
Shannon[2]	T1	0.79	0.01	0.75	0.82	NA
	T2	0.77	0.01	0.74	0.80	NA
	T3	0.76	0.01	0.73	0.79	NA
	T4	0.79	0.01	0.76	0.83	NA
	T5	0.78	0.01	0.75	0.81	NA
	T6	0.77	0.01	0.73	0.80	NA
Invsimpson[3]	T1	56.49	5.66	39.31	81.17	ab
	T2	53.70	5.28	37.61	76.66	ab
	T3	47.03	4.63	32.94	67.15	b
	T4	65.41	6.43	45.82	93.38	a
	T5	58.00	5.70	40.62	82.80	ab
	T6	58.39	6.03	40.18	84.86	ab

Continuing Table S2

Index	Sample Type	Mean	SE	lower.CL	upper.CL	Tukey HSD
Sobs[1]	BS	41.40	3.89	29.85	57.44	b
	RAL	57.63	5.17	42.17	78.77	a
	RAS	64.57	5.79	47.25	88.25	a
	RVL	54.63	4.92	39.91	74.79	a
	RVS	66.76	6.02	48.77	91.39	a
Shannon[2]	BS	0.71	0.01	0.68	0.74	b
	RAL	0.79	0.01	0.76	0.82	a
	RAS	0.80	0.01	0.77	0.82	a
	RVL	0.79	0.01	0.76	0.82	a
	RVS	0.80	0.01	0.77	0.82	a
Invsimpson[3]	BS	696.00	32.44	591.66	818.74	b
	RAL	906.79	39.95	777.74	1057.26	a
	RAS	868.49	38.26	744.91	1012.57	a
	RVL	843.50	37.39	722.78	984.37	a
	RVS	874.46	38.77	749.30	1020.53	a

Index	Sampling Time	Sample Type	Mean	SE	lower.C	upper.CL	TukeyHSD
Sobs[1]	T1	BS	76.12	13.84	40.4	143.5	a
	T1	RAL	64.71	10.45	36.9	113.6	ab
	T1	RAS	48.12	7.76	27.4	84.4	ab
	T1	RVL	63.73	10.88	35.2	115.5	ab
	T1	RVS	38.07	6.14	21.7	66.8	b
	T2	BS	35.25	5.69	20.1	61.9	b
	T2	RAL	60.28	9.74	34.3	105.8	ab
	T2	RAS	55.2	8.91	31.4	96.9	ab
	T2	RVL	58.16	9.39	33.1	102.1	ab
	T2	RVS	65.44	10.57	37.3	114.9	a
	T3	BS	18.17	2.93	10.3	31.9	c
	T3	RAL	52.29	8.44	29.8	91.8	ab
	T3	RAS	71.85	11.6	40.9	126.1	ab
	T3	RVL	41.23	6.66	23.5	72.4	b
	T3	RVS	81.79	13.21	46.6	143.6	a
	T4	BS	69.82	11.28	39.8	122.6	a
	T4	RAL	59.38	9.59	33.8	104.2	a
	T4	RAS	74.22	11.99	42.3	130.3	a
	T4	RVL	52.84	8.53	30.1	92.8	a
	T4	RVS	73.64	11.89	42	129.3	a
	T5	BS	40.56	6.55	23.1	71.2	b
	T5	RAL	50.25	8.11	28.6	88.2	ab
	T5	RAS	76.51	12.36	43.6	134.3	a
	T5	RVL	53.63	8.66	30.6	94.1	ab
	T5	RVS	78.46	12.67	44.7	137.7	a
	T6	BS	36.5	7.92	17.1	77.7	b
	T6	RAL	60.23	9.73	34.3	105.7	ab

	T6	RAS	66.89	10.8	38.1	117.4	ab
	T6	RVL	61.39	9.91	35	107.8	ab
	T6	RVS	75.19	12.83	41.5	136.3	a
Shannon[2]	T1	BS	0.8	0.02	0.7	0.9	a
	T1	RAL	0.8	0.02	0.7	0.9	a
	T1	RAS	0.77	0.02	0.7	0.8	a
	T1	RVL	0.8	0.02	0.7	0.9	a
	T1	RVS	0.76	0.02	0.7	0.8	a
	T2	BS	0.7	0.02	0.6	0.8	b
	T2	RAL	0.79	0.02	0.7	0.9	a
	T2	RAS	0.79	0.02	0.7	0.9	a
	T2	RVL	0.79	0.02	0.7	0.9	a
	T2	RVS	0.79	0.02	0.7	0.9	a
	T3	BS	0.64	0.01	0.6	0.7	b
	T3	RAL	0.79	0.02	0.7	0.8	a
	T3	RAS	0.8	0.02	0.7	0.9	a
	T3	RVL	0.77	0.02	0.7	0.8	a
	T3	RVS	0.81	0.02	0.8	0.9	a
	T4	BS	0.77	0.02	0.7	0.8	a
	T4	RAL	0.79	0.02	0.7	0.9	a
	T4	RAS	0.8	0.02	0.7	0.9	a
	T4	RVL	0.79	0.02	0.7	0.9	a
	T4	RVS	0.8	0.02	0.7	0.9	a
	T5	BS	0.71	0.02	0.7	0.8	b
	T5	RAL	0.78	0.02	0.7	0.8	a
	T5	RAS	0.81	0.02	0.7	0.9	a
	T5	RVL	0.79	0.02	0.7	0.9	a
	T5	RVS	0.81	0.02	0.7	0.9	a
	T6	BS	0.67	0.02	0.6	0.7	b
	T6	RAL	0.79	0.02	0.7	0.9	a
	T6	RAS	0.8	0.02	0.7	0.9	a
	T6	RVL	0.79	0.02	0.7	0.9	a
	T6	RVS	0.8	0.02	0.7	0.9	a
Invsimpson[3]	T1	BS	963.73	92.98	688.6	1348.8	a
	T1	RAL	906.29	77.14	673.7	1219.2	a
	T1	RAS	827.87	70.37	615.7	1113.2	a
	T1	RVL	834.35	75.06	609.8	1141.5	a
	T1	RVS	832.19	70.74	618.9	1119	a
	T2	BS	636.49	54.18	473.1	856.3	a
	T2	RAL	835.31	71.11	620.9	1123.7	a
	T2	RAS	796.51	67.8	592.1	1071.5	a
	T2	RVL	836.82	71.23	622	1125.8	a
	T2	RVS	864.07	73.55	642.3	1162.4	a
	T3	BS	419.48	35.71	311.8	564.3	b
	T3	RAL	990.54	84.32	736.3	1332.6	a
	T3	RAS	927.94	78.99	689.8	1248.3	a
	T3	RVL	837.04	71.25	622.2	1126.1	a
	T3	RVS	875.15	74.5	650.5	1177.3	a

T4	BS	859.53	73.17	638.9	1156.3	a
T4	RAL	974.79	82.98	724.6	1311.4	a
T4	RAS	922.63	78.54	685.8	1241.2	a
T4	RVL	827.32	70.43	615	1113	a
T4	RVS	895.65	76.24	665.8	1204.9	a
T5	BS	645.29	54.93	479.7	868.1	b
T5	RAL	896.89	76.35	666.7	1206.6	a
T5	RAS	915.7	77.95	680.7	1231.9	a
T5	RVL	819.16	69.73	608.9	1102	ab
T5	RVS	877.04	74.66	651.9	1179.9	ab
T6	BS	796.47	92.31	531.9	1192.7	a
T6	RAL	848.02	72.19	630.4	1140.8	a
T6	RAS	830.1	70.66	617	1116.7	a
T6	RVL	909.35	77.41	675.9	1223.3	a
T6	RVS	904.55	81.56	660.7	1238.5	a

[1]Number of observed OTUs, [2]Shannon's evenness; [3]inverse of the Simpson diversity index; [4]Sampling was performed every 2 hours over the course of 10 hours, starting 1 hour prior to morning feeding (~ 9 AM) and ending just prior to evening feeding (~ 7 PM), totaling six time points (T1-T6). [5]BS: buccal swab, RAL: rumen anterior liquid, RAS: rumen anterior solid, RVL: rumen ventral liquid and RVS: rumen ventral solid; [6] Interaction terms: Sampling time vs Sample type. Means between groups in the row (class, time or interaction sample type within sampling time) followed by the same letter are not significantly different ($P > 0.05$) by Tukey HSD test. NA: not available, post hoc was not performed due to lack of overall statistically significant.

TABLE S3. Beta diversity analysis: summary of the permutational multivariate analysis of variance (PERMANOVA nperm=1000) of Bray-Curtis dissimilarities in the bacterial community.

PERMANOVA- adonis model					
Factor	**Df**	**Sum Of Sqs**	**R^2**	**F**	**Pr(>F)**
Time	5	1.724	0.04462	3.1937	0.000999
Sample type	4	10.986	0.28425	25.4349	0.000999
Time*Sample type	20	4.126	0.10675	1.9104	0.000999
Residual	202	21.811	0.56438		
Total	231	38.647	1.00000		

PERMANOVA- pairwise adonis					
Pairs: Sample Type[1]	**SumsOfSqs**	**F.Model**	**R^2**	**P-value**	**P-adjusted***
BS vs RAL	5.742	31.734	0.265	0.001	0.010
BS vs RAS	5.064	28.353	0.244	0.001	0.010
BS vs RVL	5.821	32.218	0.270	0.001	0.010
BS vs RVS	5.049	28.222	0.245	0.001	0.010
RAL vs RAS	1.634	19.198	0.170	0.001	0.010
RAL vs RVL	0.045	0.522	0.006	0.916	1.000
RAL vs RVS	1.706	20.210	0.179	0.001	0.010
RAS vs RVL	1.706	20.346	0.180	0.001	0.010
RAS vs RVS	0.029	0.348	0.004	0.987	1.000
RVL vs RVS	1.764	21.228	0.187	0.001	0.010
Pairs: Time[2]					
T1 vs T2	0.236	1.524	0.020	0.135	0.208
T1 vs T3	0.950	5.530	0.069	0.001	0.007
T1 vs T4	0.314	2.496	0.032	0.005	0.019
T1 vs T5	0.526	3.477	0.044	0.004	0.019
T1 vs T6	0.204	1.692	0.024	0.061	0.131
T2 vs T3	0.353	1.713	0.021	0.139	0.208
T2 vs T4	0.247	1.524	0.019	0.158	0.215
T2 vs T5	0.217	1.165	0.015	0.252	0.315
T2 vs T6	0.286	1.790	0.024	0.058	0.131
T3 vs T4	0.467	2.613	0.032	0.039	0.117
T3 vs T5	0.178	0.879	0.011	0.415	0.478
T3 vs T6	0.709	4.002	0.052	0.001	0.007
T4 vs T5	0.134	0.847	0.011	0.514	0.551
T4 vs T6	0.102	0.789	0.011	0.694	0.694
T5 vs T6	0.256	1.642	0.022	0.102	0.191
Pairs: Sample type within Time					
T1:BS vs T1:RAL	0.422	3.902	0.245	0.001	0.002
T1:BS vs T1:RAS	0.199	1.775	0.129	0.043	0.054
T1:BS vs T1:RVL	0.417	3.674	0.250	0.001	0.002
T1:BS vs T1:RVS	0.229	2.098	0.149	0.013	0.017
T1:RAL vs T1:RAS	0.415	4.982	0.262	0.001	0.002
T1:RAL vs T1:RVL	0.021	0.250	0.019	0.987	0.994
T1:RAL vs T1:RVS	0.456	5.663	0.288	0.001	0.002
T1:RAS vs T1:RVL	0.426	4.973	0.277	0.001	0.002
T1:RAS vs T1:RVS	0.018	0.217	0.015	0.990	0.994
T1:RVL vs T1:RVS	0.463	5.587	0.301	0.001	0.002
T2:BS vs T2:RAL	1.268	5.990	0.300	0.002	0.003
T2:BS vs T2:RAS	1.187	5.757	0.291	0.002	0.003
T2:BS vs T2:RVL	1.287	6.094	0.303	0.002	0.003
T2:BS vs T2:RVS	1.188	5.776	0.292	0.001	0.002

T2:RAL vs T2:RAS	0.371	4.092	0.226	0.001	0.002
T2:RAL vs T2:RVL	0.013	0.135	0.010	0.994	0.994
T2:RAL vs T2:RVS	0.361	4.001	0.222	0.001	0.002
T2:RAS vs T2:RVL	0.388	4.304	0.235	0.002	0.003
T2:RAS vs T2:RVS	0.018	0.212	0.015	0.989	0.994
T2:RVL vs T2:RVS	0.376	4.188	0.230	0.001	0.002
T3:BS vs T3:RAL	3.181	43.392	0.756	0.002	0.003
T3:BS vs T3:RAS	3.200	42.116	0.751	0.001	0.002
T3:BS vs T3:RVL	3.218	43.413	0.756	0.001	0.002
T3:BS vs T3:RVS	3.247	46.646	0.769	0.001	0.002
T3:RAL vs T3:RAS	0.473	5.530	0.283	0.001	0.002
T3:RAL vs T3:RVL	0.038	0.451	0.031	0.935	0.994
T3:RAL vs T3:RVS	0.467	5.910	0.297	0.001	0.002
T3:RAS vs T3:RVL	0.488	5.651	0.288	0.002	0.003
T3:RAS vs T3:RVS	0.026	0.319	0.022	0.993	0.994
T3:RVL vs T3:RVS	0.475	5.942	0.298	0.001	0.002
T4:BS vs T4:RAL	0.592	3.557	0.203	0.001	0.002
T4:BS vs T4:RAS	0.448	2.758	0.165	0.001	0.002
T4:BS vs T4:RVL	0.583	3.461	0.198	0.002	0.003
T4:BS vs T4:RVS	0.445	2.679	0.161	0.001	0.002
T4:RAL vs T4:RAS	0.255	3.375	0.194	0.001	0.002
T4:RAL vs T4:RVL	0.027	0.331	0.023	0.980	0.994
T4:RAL vs T4:RVS	0.289	3.650	0.207	0.001	0.002
T4:RAS vs T4:RVL	0.244	3.138	0.183	0.001	0.002
T4:RAS vs T4:RVS	0.024	0.321	0.022	0.990	0.994
T4:RVL vs T4:RVS	0.269	3.296	0.191	0.002	0.003
T5:BS vs T5:RAL	1.551	8.337	0.373	0.001	0.002
T5:BS vs T5:RAS	1.494	8.381	0.374	0.001	0.002
T5:BS vs T5:RVL	1.629	9.283	0.399	0.001	0.002
T5:BS vs T5:RVS	1.495	8.441	0.376	0.001	0.002
T5:RAL vs T5:RAS	0.271	3.172	0.185	0.001	0.002
T5:RAL vs T5:RVL	0.024	0.286	0.020	0.988	0.994
T5:RAL vs T5:RVS	0.310	3.686	0.208	0.001	0.002
T5:RAS vs T5:RVL	0.290	3.877	0.217	0.001	0.002
T5:RAS vs T5:RVS	0.021	0.276	0.019	0.992	0.994
T5:RVL vs T5:RVS	0.326	4.420	0.240	0.001	0.002
T6:BS vs T6:RAL	0.554	3.862	0.279	0.003	0.004
T6:BS vs T6:RAS	0.478	3.625	0.266	0.004	0.005
T6:BS vs T6:RVL	0.572	4.094	0.290	0.003	0.004
T6:BS vs T6:RVS	0.432	3.063	0.254	0.003	0.004
T6:RAL vs T6:RAS	0.231	2.682	0.161	0.001	0.002
T6:RAL vs T6:RVL	0.036	0.390	0.027	0.965	0.994
T6:RAL vs T6:RVS	0.221	2.477	0.160	0.002	0.003
T6:RAS vs T6:RVL	0.276	3.308	0.191	0.001	0.002
T6:RAS vs T6:RVS	0.027	0.335	0.025	0.978	0.994
T6:RVL vs T6:RVS	0.249	2.887	0.182	0.001	0.002

BS: buccal swab, RAL: rumen anterior liquid, RAS: rumen anterior solid, RVL: rumen ventral liquid and RVS: rumen ventral solid; [2]Sampling was performed every 2 hours over the course of 10 hours, starting 1 hour prior to morning feeding (~ 9 AM) and ending just prior to evening feeding (~ 7 PM), totaling six time points (T1-T6); *P-value adjusted by Benjamini-Hochberg correction method.

TABLE S4. Performance of Random Forest (5 classes) according to tuning methods

Method	Sample[2]	Sensitivity	Specificity	Precision	Recall	F1	B. Accuracy[3]
Default[1]	BS	1.00	1.00	1.00	1.00	1.00	1.00
	RAL	0.60	0.89	0.60	0.60	0.60	0.75
	RAS	0.27	0.83	0.23	0.27	0.25	0.55
	RVL	0.54	0.89	0.54	0.54	0.54	0.72
	RVS	0.33	0.85	0.38	0.33	0.36	0.59
Random	BS	0.88	1.00	1.00	0.88	0.93	0.94
	RAL	0.67	0.84	0.53	0.67	0.59	0.75
	RAS	0.55	0.83	0.38	0.55	0.44	0.69
	RVL	0.31	0.91	0.44	0.31	0.36	0.61
	RVS	0.47	0.91	0.58	0.47	0.52	0.69
Ranger	BS	1.00	1.00	1.00	1.00	1.00	1.00
	RAL	0.47	0.84	0.44	0.47	0.45	0.65
	RAS	0.45	0.83	0.33	0.45	0.38	0.64
	RVL	0.31	0.86	0.33	0.31	0.32	0.58
	RVS	0.33	0.89	0.45	0.33	0.38	0.61
Gridsearch	BS	0.94	1.00	1.00	0.94	0.97	0.97
	RAL	0.53	0.82	0.44	0.53	0.48	0.68
	RAS	0.55	0.83	0.38	0.55	0.44	0.69
	RVL	0.23	0.88	0.30	0.23	0.26	0.55
	RVS	0.40	0.91	0.55	0.40	0.46	0.65

[1]Non tuning was applied; [2]BS: buccal swab, RAL: rumen anterior liquid, RAS: rumen anterior solid, RVL: rumen ventral liquid and RVS: rumen ventral solid; [3]Balanced accuracy.

TABLE S5. Performance of Random Forest (3 classes) by sample type using different re-sampling methods

Method	Sample[5]	Sensitivity	Specificity	Precision	Recall	F1	B. Accuracy
Default[1]	BS	0.833	1.000	1.000	0.833	0.909	0.917
	RL	1.000	1.000	1.000	1.000	1.000	1.000
	RS	1.000	0.950	0.933	1.000	0.966	0.975
Up[2]	BS	0.833	1.000	1.000	0.833	0.909	0.917
	RL	1.000	1.000	1.000	1.000	1.000	1.000
	RS	1.000	0.950	0.933	1.000	0.966	0.975
Down[3]	BS	0.833	1.000	1.000	0.833	0.909	0.917
	RL	1.000	1.000	1.000	1.000	1.000	1.000
	RS	1.000	0.950	0.933	1.000	0.966	0.975
SMOTE[4]	BS	0.917	1.000	1.000	0.917	0.957	0.958
	RL	1.000	1.000	1.000	1.000	1.000	1.000
	RS	1.000	0.975	0.966	1.000	0.982	0.988

[1]Non re-sampling was applied; [2]over-sampling of the minority class, [3]under-sampling of the majority class, [4]Synthetic Minority Oversampling Technique: combination of up and down re-sampling methods (Chawla et al., 2002. https://doi.org/10.1613/jair.953); [5]BS= buccal swab, rumen samples were merged based on rumen content strata: RL=rumen liquids (RAL +RVL) and RS=rumen solids (RAS+RVS).

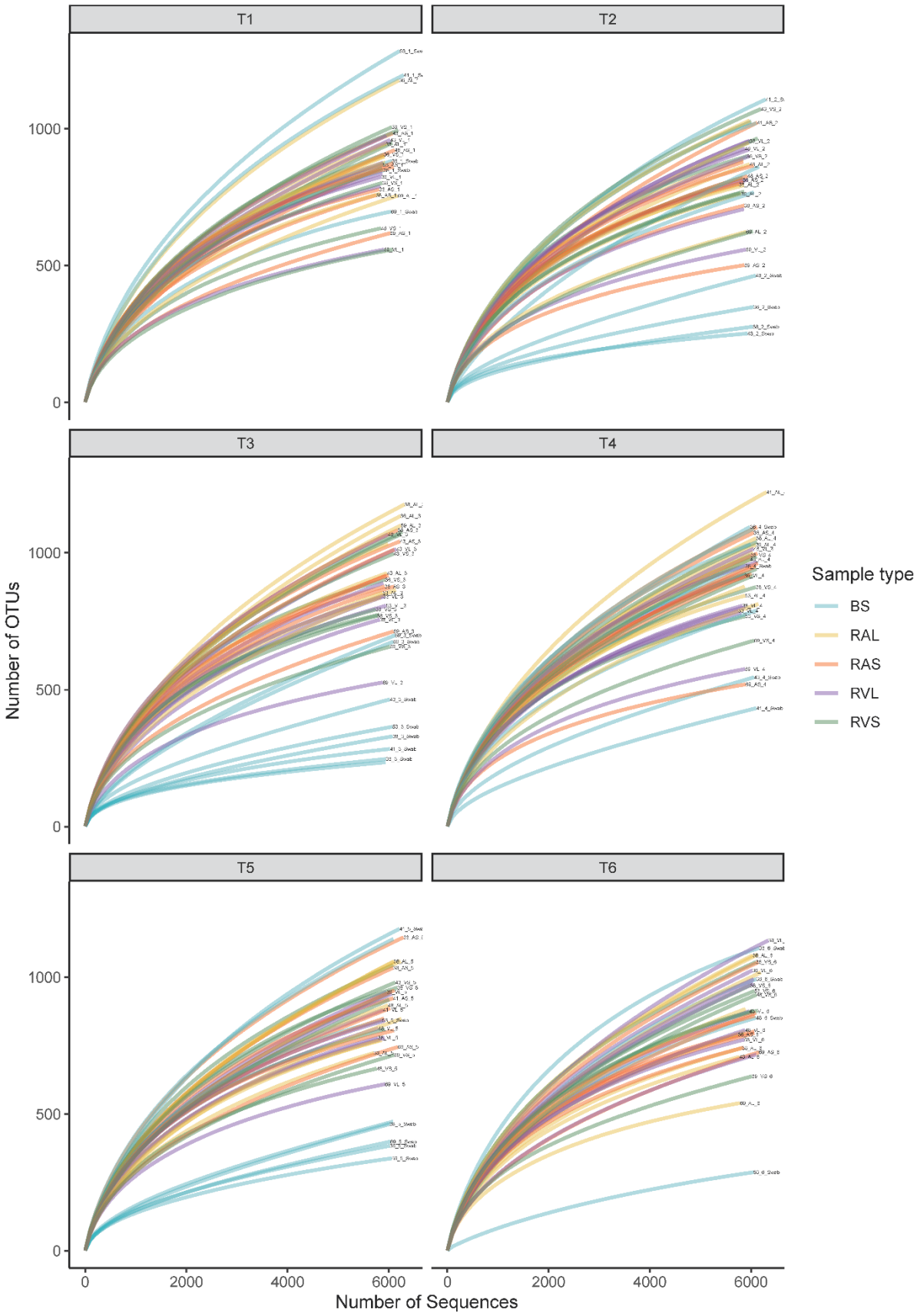
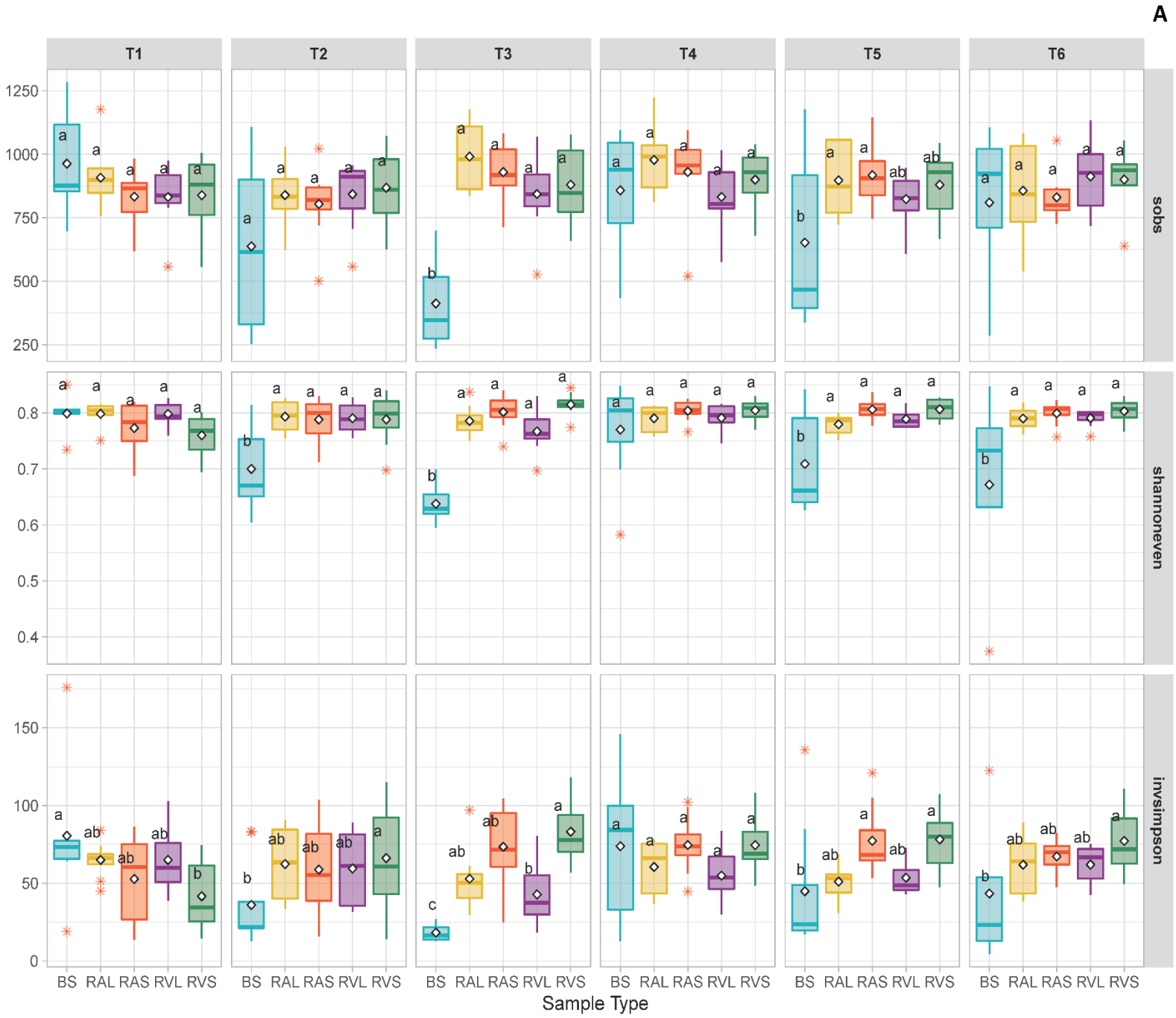


FIG S1. Rarefaction curves based on observed species (Number of OTUs) showing the depth of coverage (Number of Sequences) for each sample grouped according to sample type (BS: buccal swab, RAL: rumen anterior liquid, RAS: rumen anterior solid, RVL: rumen ventral liquid and RVS: rumen ventral solid) in each facet that represents

sampling time (T1-T6). Sampling was performed every 2 hours over the course of 10 hours, starting 1 hour prior to morning feeding (T1=~ 9 AM) and ending just prior to evening feeding (T6=~ 7 PM).

FIG S2. Distribution of the most abundant bacterial taxa (OTUs summarized at phylum **(A)**, family **(B)**, and genus **(C)** according to sample type (BS: buccal swab, RAL: rumen anterior liquid, RAS: rumen anterior solid, RVL: rumen

ventral liquid and RVS: rumen ventral solid) and sampling time (T1-T6). Sampling was performed every 2 hours over the course of 10 hours, starting 1 hour prior to morning feeding (T1=~ 9 AM) and ending just prior to evening feeding (T6=~ 7 PM).


A
T1
T2
T3
T4
T5
T6
sobs
shannoneven
invsimpson
1250
1000
750
500
250
0.8
0.7
0.6
0.5
0.4
150
100
50
0
BS RAL RAS RVL RVS
Sample Type

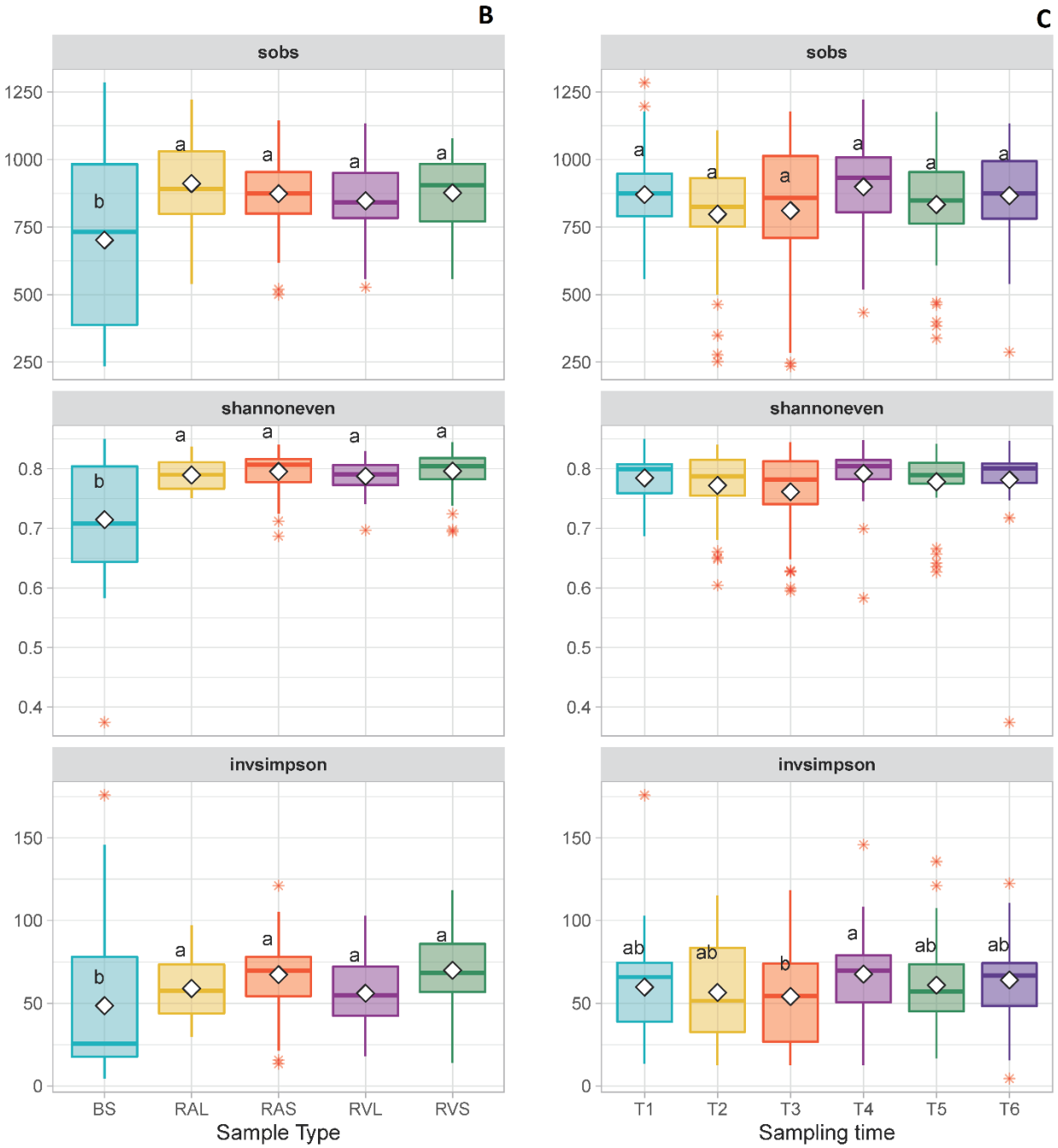


FIG S3. Alpha diversity of bacterial communities. (A) Boxplots demonstrate the distribution of alpha diversity indices: Number of observed OTUs (Sobs), Shannon's evenness (shannoneven) and inverse of Simpson's diversity (Invsimpson) for each sample type (BS: buccal swab, RAL: rumen anterior liquid, RAS: rumen anterior solid, RVL: rumen ventral liquid and RVS: rumen ventral solid) within each sampling time (T1 to T6). Plots B and C demonstrate the distribution of alpha diversity indices per sample and sampling time, respectively. Boxes represent the interquartile range (IQR) between the first(25^{th}) and third quartiles (75th percentiles) whereas the horizontal line represents the median and the white diamond represents the mean. Whiskers represent the lowest and highest values within 1.5 times the IQR from the first and third quartiles, respectively and the "*" represent outliers. LSmeans (back-transformed from the log scale) between groups (sample type within each sampling time, sample type and sampling time) followed by the same letter are not significantly different ($P > 0.05$) by Tukey HSD test.

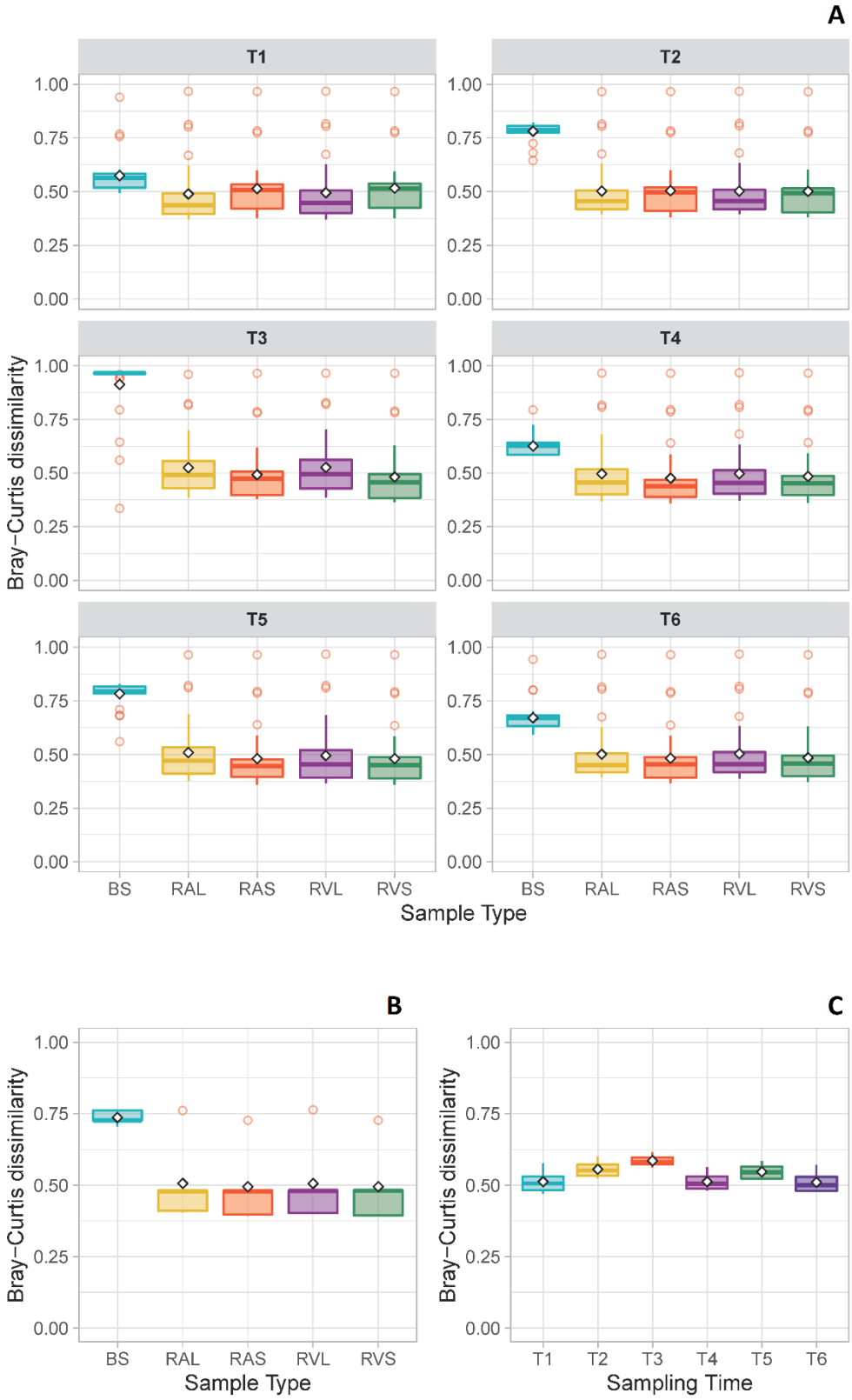


FIG S4. Beta diversity of bacterial communities. (A) Boxplots demonstrate the distribution of Bray-Curtis dissimilarity for each sample type (BS: buccal swab, RAL: rumen anterior liquid, RAS: rumen anterior solid, RVL: rumen ventral liquid and RVS: rumen ventral solid) within each sampling time (T1 to T6). Plots B and C demonstrate the distribution of Bray-Curtis dissimilarity per sample and sampling time, respectively. Boxes represent the interquartile range (IQR) between the first(25^{th}) and third quartiles (75th percentiles) whereas the horizontal line represents the median and the white diamond represents the mean.

Whiskers represent the lowest and highest values within 1.5 times the IQR from the first and third quartiles, respectively and the "o" represent outliers.

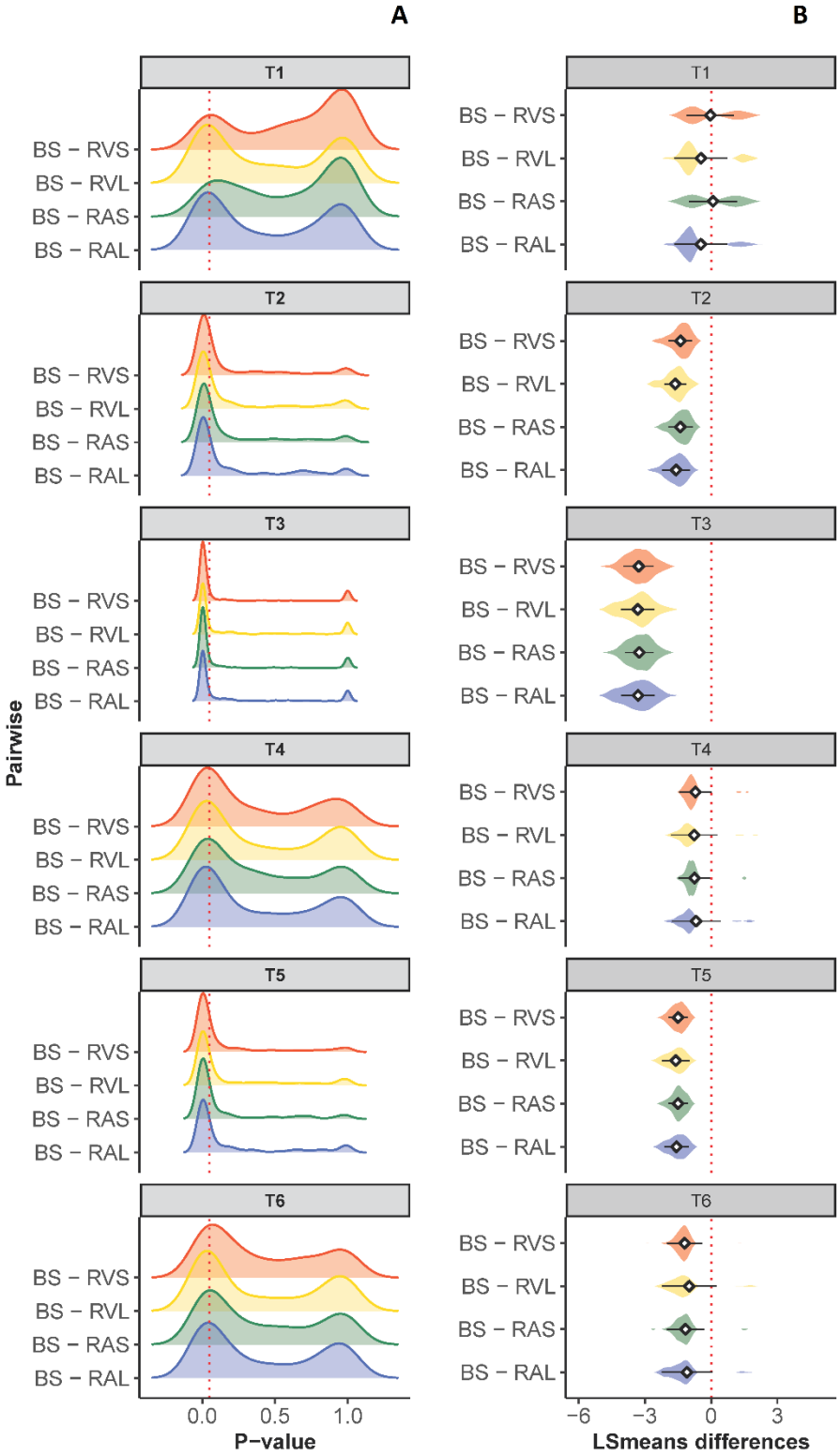


FIG S5. A) Ridgeline plots showing the distribution of bacterial OTUs whose abundance varied significantly (red line = P-value ≤ 0.05) in pairwise comparisons between buccal swab (BS) and all types of rumen samples (RAL, RVL, RAS, and RVS) within each sampling time (T1:T6). **B)** Violin plot showing the Least Squares Means (LSmeans) differences of significant pairwise comparisons (Tukey HSD ≤ 0.05) between buccal swab and all types of rumen samples within each sampling time. The white diamond and whiskers represent the mean and standard deviation, respectively.

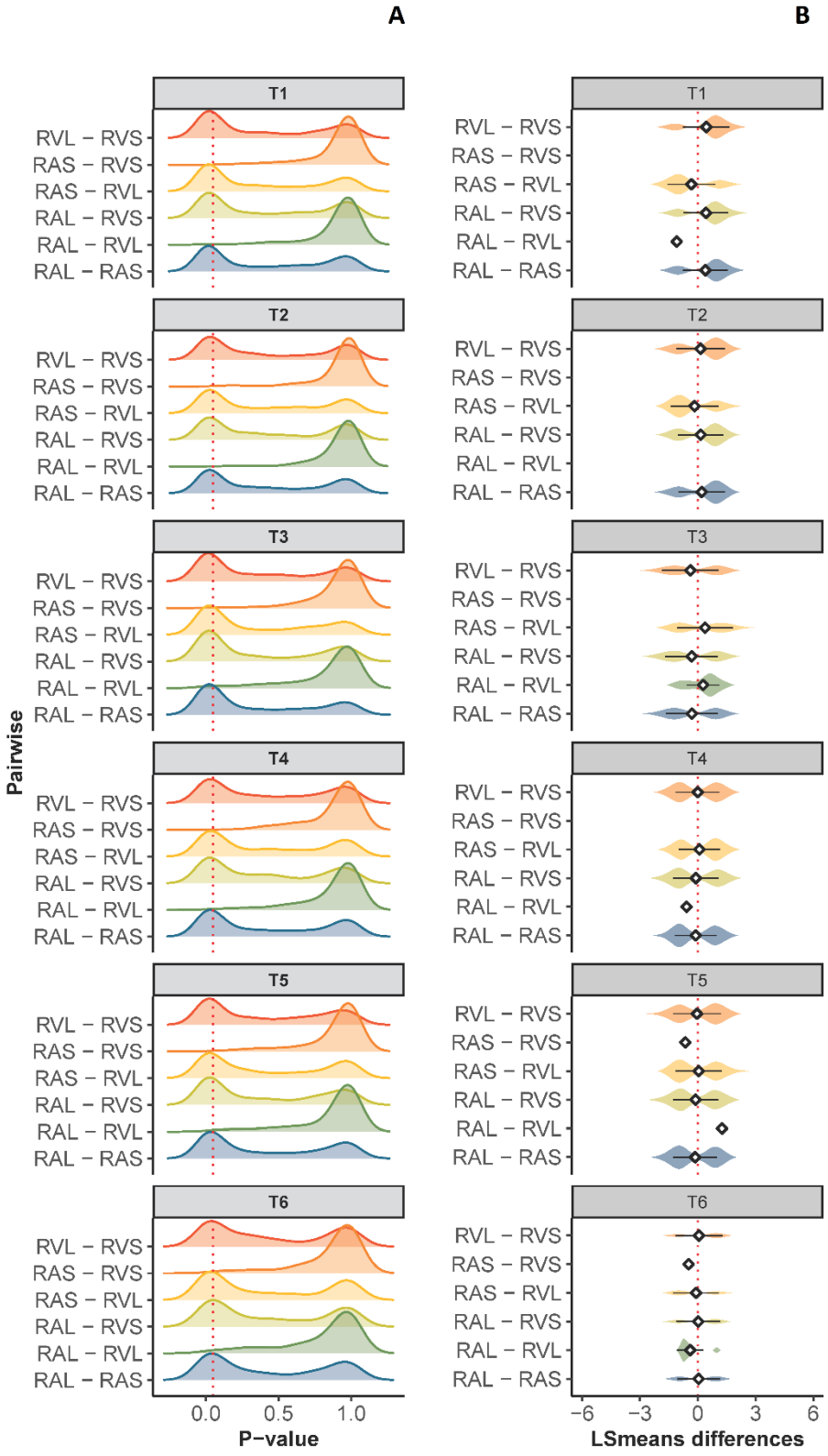


FIG S6. A) Ridgeline plot showing the distribution of bacterial OTUs whose abundance varied significantly (red line=P-value≤0.05) in pairwise comparisons between all types of rumen samples (RAL, RVL, RAS, and RVS) within each sampling time (T1:T6). **B)** Violin plot showing Least Square Means (LSMEANS) differences of significant pairwise comparisons (Tukey HSD ≤0.05) between all types of rumen samples within each sampling time.

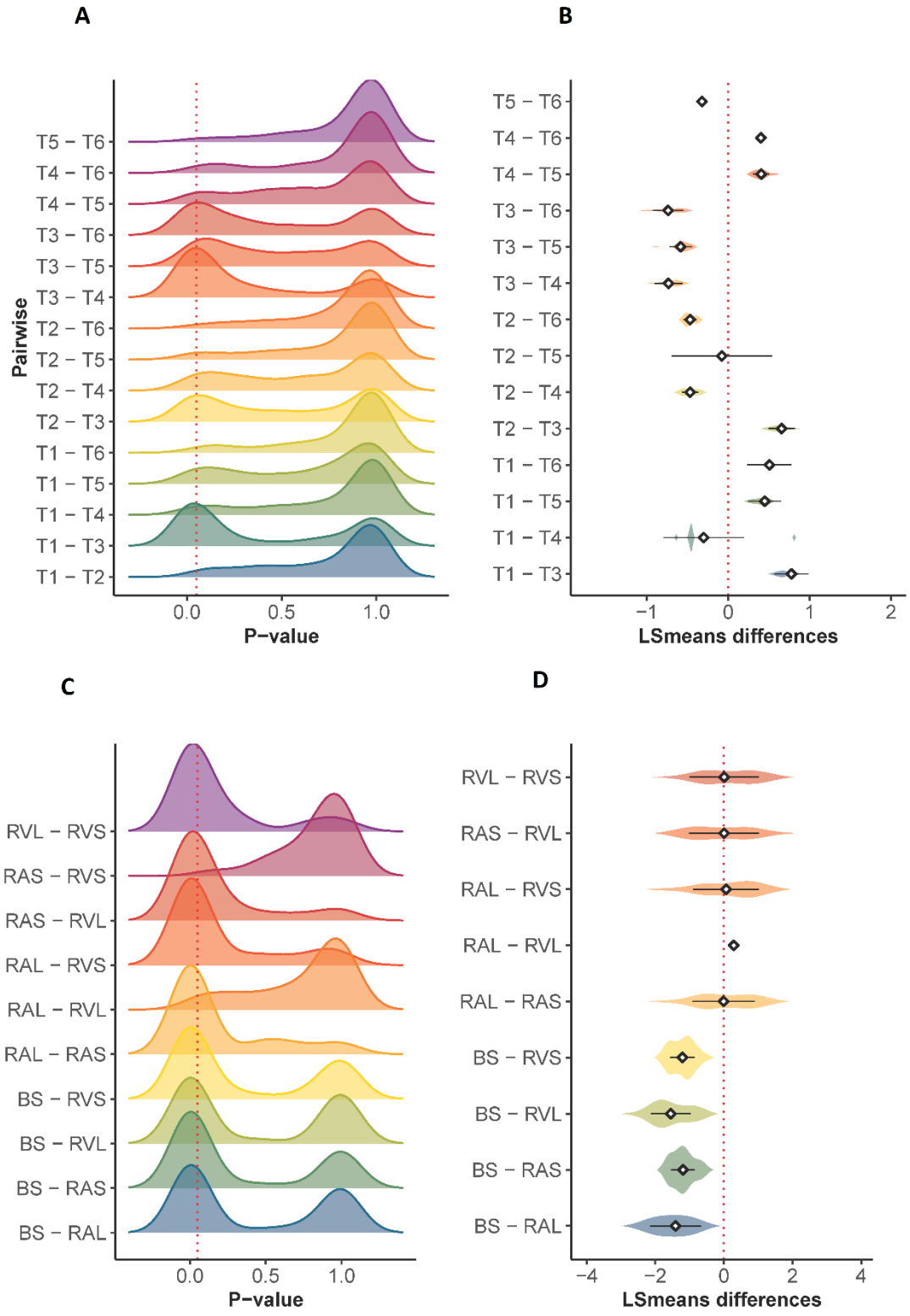


FIG S7. Ridgeline plots showing the distribution of bacterial OTUs whose abundance varied significantly (red line=P-value≤0.05) in pairwise comparisons between all the sampling times=T1-T6 (**A**) and between all the samples types=BS: buccal swab, RAL: rumen anterior liquid, RAS: rumen anterior solid, RVL: rumen ventral liquid and RVS: rumen ventral solid (**C**)**.** Violin plot showing the Least Square Means (LSmeans) differences of significant pairwise comparisons between all sampling time **(B)** and all sample types **(D)** and. Sampling was performed every 2 hours over the course of 10 hours, starting 1 hour prior to morning feeding (T1=~ 9 AM) and ending just prior to evening feeding (T6=~ 7 PM). Pairwise comparison between RAS-RVS were not significant and thus, not shown in the violin plot.

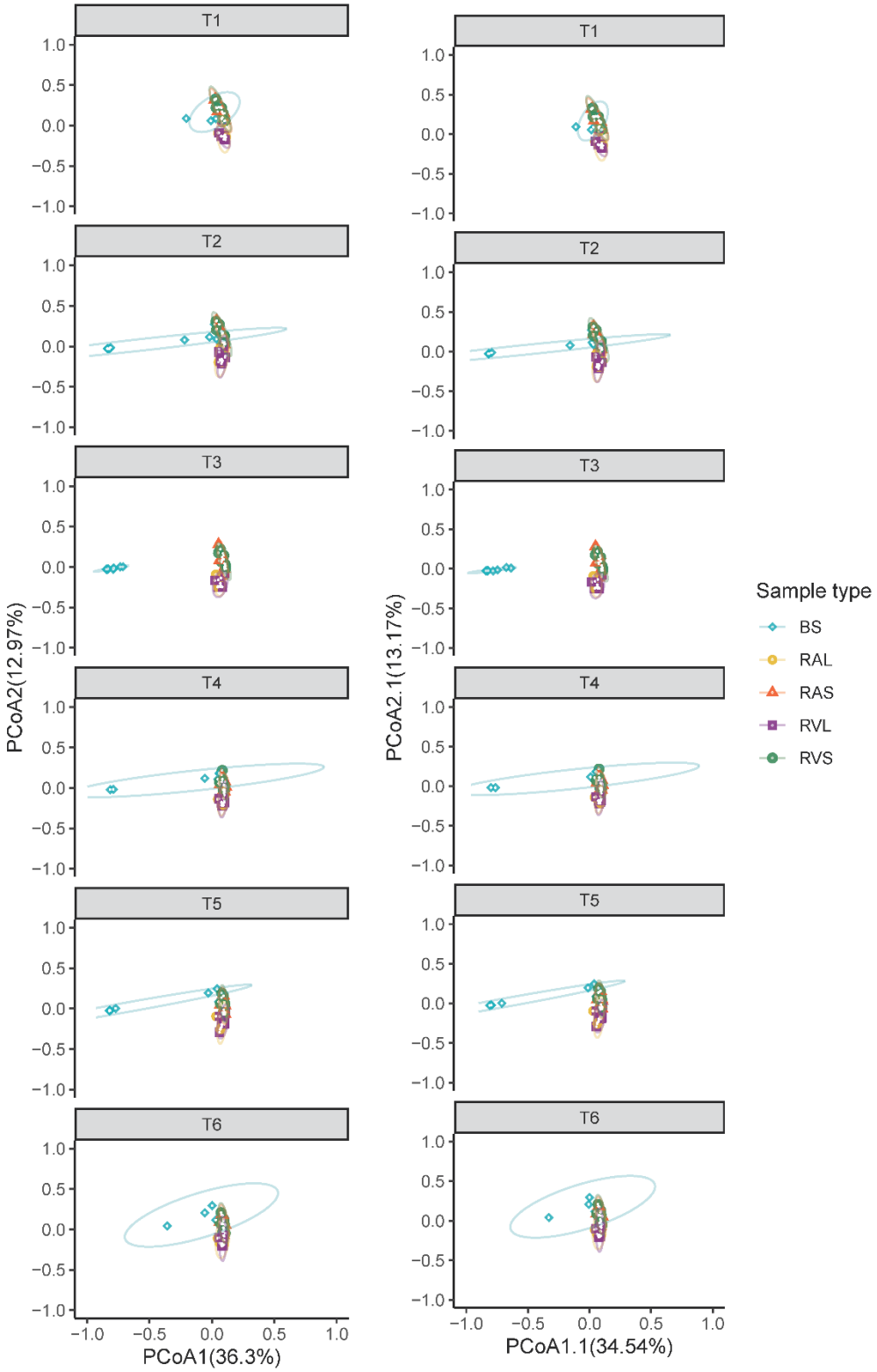


FIG S8. Principal coordinate analysis (PCoA) showing Bray-Curtis dissimilarities in the composition of bacterial communities between sample types within each sampling time. **A)** PCoA performed with 8,147 unique OTUs prior to the exclusion of potential oral taxa identified using Random Forest. **B)** PCoA performed with 8,110 unique OTUs remaining after the exclusion of 37 potential oral taxa identified using Random Forest algorithm and that presented Variance Importance (VIMP) score>=10% (please see Figure 4). Individual points in each plot represent a dairy cow, different colors and shapes represent a sample type (BS: buccal swab, RAL: rumen anterior liquid, RAS: rumen anterior solid, RVL: rumen ventral liquid and RVS: rumen ventral solid), and each facet represents a time point (T1 to T6). Percentages showed along the axes represent, respectively, the proportion of dissimilarities captured by PCoA in 2D coordinate space.

#Mothur analysis

```
make.contigs(file=RumenTimecourse2.txt, processors=10)

summary.seqs(fasta=RumenTimecourse2.trim.contigs.fasta)

screen.seqs(fasta=RumenTimecourse2.trim.contigs.fasta,
group=RumenTimecourse2.contigs.groups, maxambig=0, maxhomop=8,
maxlength=300)

unique.seqs(fasta=RumenTimecourse2.trim.contigs.good.fasta)

count.seqs(name=RumenTimecourse2.trim.contigs.good.names,
group=RumenTimecourse2.contigs.good.groups)

summary.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.fasta,
count=RumenTimecourse2.trim.contigs.good.count_table)

align.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.fasta,
reference=/home/GLBRCORG/jhskarlupka/SequencingProjects/MothurFiles/sil
va.nr_v132.align, flip=T)

summary.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.align,
count=RumenTimecourse2.trim.contigs.good.count_table)

screen.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.align,
count=RumenTimecourse2.trim.contigs.good.count_table,
summary=RumenTimecourse2.trim.contigs.good.unique.summary, start=13862,
end=23444)

summary.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.good.align
, count=RumenTimecourse2.trim.contigs.good.good.count_table)

filter.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.good.align,
vertical=T, trump=.)

unique.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.good.filter
.fasta, count=RumenTimecourse2.trim.contigs.good.good.count_table)

pre.cluster(fasta=RumenTimecourse2.trim.contigs.good.unique.good.filter
.unique.fasta,
count=RumenTimecourse2.trim.contigs.good.unique.good.filter.count_table
, diffs=2)

summary.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.good.filte
r.unique.precluster.fasta,
count=RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.prec
luster.count_table)

chimera.uchime(fasta=RumenTimecourse2.trim.contigs.good.unique.good.fil
ter.unique.precluster.fasta,
count=RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.prec
luster.count_table, dereplicate=t)

remove.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.good.filter
.unique.precluster.fasta,
count=RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.prec
```

```
luster.count_table,
accnos=RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.pre
cluster.denovo.uchime.accnos)

summary.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.good.filte
r.unique.precluster.pick.fasta,
count=RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.prec
luster.pick.count_table)

classify.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.good.filt
er.unique.precluster.pick.fasta,
count=RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.prec
luster.pick.count_table,
reference=/home/GLBRCORG/jhskarlupka/SequencingProjects/MothurFiles/sil
va.nr_v132.align,
taxonomy=/home/GLBRCORG/jhskarlupka/SequencingProjects/MothurFiles/silv
a.nr_v132.tax, cutoff=80)

remove.lineage(fasta=RumenTimecourse2.trim.contigs.good.unique.good.fil
ter.unique.precluster.pick.fasta,
count=RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.prec
luster.pick.count_table,
taxonomy=RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.p
recluster.pick.nr_v132.wang.taxonomy, taxon=unknown;-Archaea;-
Eukaryota;-Bacteria;Cyanobacteria;Chloroplast;-
Bacteria;Proteobacteria;Alphaproteobacteria;Rickettsiales;mitochondria;
)

summary.seqs(fasta=RumenTimecourse2.trim.contigs.good.unique.good.filte
r.unique.precluster.pick.pick.fasta,
count=RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.prec
luster.pick.pick.count_table)

system(cp
RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.precluster
.pick.pick.fasta RumenTimecourse2.final.fasta)

system(cp
RumenTimecourse2.trim.contigs.good.unique.good.filter.unique.precluster
.pick.pick.count_table RumenTimecourse2.final.count_table)

count.groups(count=RumenTimecourse2.final.count_table)

summary.seqs(fasta=RumenTimecourse2.final.fasta,
count=RumenTimecourse2.final.count_table)

split.abund(fasta=RumenTimecourse2.final.fasta,
count=RumenTimecourse2.final.count_table, cutoff=1)

summary.seqs(fasta=RumenTimecourse2.final.abund.fasta,
count=RumenTimecourse2.final.abund.count_table)

dist.seqs(fasta=RumenTimecourse2.final.abund.fasta)
```

```
cluster.split(column=RumenTimecourse2.final.abund.dist,
count=RumenTimecourse2.final.abund.count_table, method=opti,
cutoff=0.03)

make.shared(list=RumenTimecourse2.final.abund.opti_mcc.unique_list.list
count=RumenTimecourse2.final.abund.count_table, label=0.03)

classify.seqs(fasta=RumenTimecourse2.final.abund.fasta,
count=RumenTimecourse2.final.abund.count_table, template=/home/
MothurFiles/silva.nr_v132.align,taxonomy=/home/MothurFiles/silva.nr_v13
2.tax, cutoff=80)

classify.otu(list=RumenTimecourse2.final.abund.opti_mcc.list,
taxonomy=RumenTimecourse2.final.abund.nr_v132.wang.taxonomy,
count=RumenTimecourse2.final.abund.count_table, label=0.03, cutoff=80,
basis=otu, probs=F)

summary.single(shared=RumenTimecourse2.final.abund.opti_mcc.shared,
label=0.03, calc=nseqs-sobs-coverage)
```

#R analyses

```
#Summary Sequencing (TABLE S1): Summary of sequencing and alpha
diversity indices)
rm(list=ls())
library(data.table)
library(reshape)
library(plyr)

#load
coverage<- read.table(file="RumenTimecourse_PreNorm.groups.summary",
header=T)
alpha<- read.table(file="RumenTimecourse_Norm.groups.summary",
header=T)
meta_data=read.table(file="meta_data_Joe.txt", header=TRUE, sep = "\t")

#merge/melt
merge=merge(meta_data, alpha, by="group")
subset=subset(merge, select= c("group", "cow", "class", "time",
"nseqs", "coverage",
"sobs", "ace", "shannoneven", "simpsoneven","bergerparker",
"invsimpson"))


melt=melt(subset, id.vars=c("cow", "group", "class", "time"))

#summary by time and sample type
se <- function(x) {sd(x)/sqrt(length(x))}
summary1=cast(melt, class+time~variable, value='value',fun.aggregate =
c(length, sum, mean, sd))
summary1 =data.frame(lapply(summary1, function(y) if(is.numeric(y))
formatC (y, 2, format="f") else y))


#export
library(openxlsx)
write.xlsx(summary1,
file="code/summary_sequencing_alpha_diversity.xlsx")
```

#Anova Alpha diversity

```
rm(list=ls())
library(MASS)
library(multcomp)
library(nlme)
library(lsmeans)
library(ggplot2)

library(data.table)
#library(reshape)
#library(plyr)

#load
alpha<- read.table(file="RumenTimecourse_Norm.groups.summary",
header=T)
meta_data=read.table(file="meta_data_Joe.txt", header=TRUE, sep = "\t")

#merge/melt
merge=merge(meta_data, alpha, by="group")
subset=subset(merge, select= c("group", "cow", "class", "time",
"sobs", "ace", "shannoneven", "simpsoneven","bergerparker",
"invsimpson"))

dat=subset

resp_alpha <- names(dat)[!(names(dat) %in% c("cow","time", "class",
"group"))]



dat$cow=as.factor(dat$cow)
dat$group=as.factor(dat$group)
dat$time=as.factor(dat$time)
dat$class=as.factor(dat$class)


set.seed(12)
analise<- lsm_dtime <- lsm_dclass <- lsm_dint <- cld_dtime <-cld_dclass <-
cld_dint<- contr_dtime <- contr_dclass <- contr_dint <-NULL

for(i in resp_alpha){
bac<-data.frame(dat[2:4],y=dat[,i])
fit<-glmmPQL (y~time*class, random=~1|cow/time, family=Gamma(link = log),
correlation=corAR1(), data = bac)
anova=anova.lme(fit, type="sequential", adjustSigma=FALSE)
analise<-rbind(analise,data.frame("alpha_index"=i,anova))

lsm_time <- lsmeans(fit, ~ time, type="response")
lsm_dtime<-rbind(lsm_dtime,data.frame("alpha_index"=i, lsm_time))
cld_time <- cld(lsm_time, Letters=letters, reverse=T, alpha=0.05,
type="response", adjust="tukey")
cld_dtime<- rbind(cld_dtime,data.frame("alpha_index"=i, cld_time))
```

```
lsm_class <- lsmeans(fit, ~ class, type="response")
lsm_dclass<-rbind(lsm_dclass,data.frame("alpha_index"=i, lsm_class))
cld_class <- cld(lsm_class, Letters=letters, reverse=T, alpha=0.05,
type="response", adjust="tukey")
cld_dclass<- rbind(cld_dclass,data.frame("alpha_index"=i, cld_class))

lsm_int <- lsmeans(fit, ~ class|time, type="response")
lsm_dint<-rbind(lsm_dint,data.frame("alpha_index"=i, lsm_int))
cld_int <- cld(lsm_int, Letters=letters, reverse=T, alpha=0.05,
type="response", adjust="tukey")
cld_dint<- rbind(cld_dint,data.frame("alpha_index"=i, cld_int))
}

#Organizing to export
analise$factor=rownames(analise)
rownames(analise)=NULL
analise=analise[c(5,6,1:4)]


library(openxlsx)
results <- list("Anova_Gamma_distribuition" = analise, "Tukey_letters_time" =
cld_dtime, "Tukey_letters_class"= cld_dclass, "Tukey_letters_int"= cld_dint)

write.xlsx(results, file = "code/Anova_Alpha_Results.xlsx")


#plot
library(ggplot2)
library(data.table)
#int
data=melt(subset, id.vars=c("cow", "group", "class", "time"))
colnames(data)[5]=c("alpha_index")
data_int=merge(data, cld_dint, by=c("alpha_index","class","time"))
selected=c("sobs","shannoneven","invsimpson")
data_int= data_int[data_int$alpha_index%in%selected,]
data_int$alpha_index =droplevels(data_int$alpha_index)

pdf("alpha_boxplot_int.pdf", useDingbats=FALSE, onefile=T, paper='A4r',
width=12, height=7)
ggplot(data_int, aes(x = class, y = value)) +
geom_boxplot(aes(fill=class, colour=class), alpha=0.4,outlier.colour =
"red", outlier.shape = 8) + facet_grid(alpha_index~time,
scales="free_y")+ scale_fill_manual(values = c("#00AFBB", "#E7B800",
"#FC4E07", "magenta4", "seagreen")) +scale_colour_manual(values=
c("#00AFBB", "#E7B800", "#FC4E07", "magenta4", "seagreen"))+
theme_light()+theme(legend.position="none")  + theme(strip.background =
element_rect(fill="grey85"))+ theme(strip.text.x = element_text(colour
= "black", face = "bold"), strip.text.y = element_text(colour =
"black", face = "bold")) +labs(x = "Sample Type",y = "",color = "")
```

```
+geom_text(data=data_int, aes(x=class, y=response, label=.group), vjust=-2,
hjust=1, check_overlap = T, show.legend = FALSE, size=3)

dev.off()

#class
data_class=merge(data, cld_dclass, by=c("alpha_index","class"))
selected=c("sobs","shannoneven","invsimpson")
data_class= data_class[data_class$alpha_index%in%selected,]
data_class$alpha_index =droplevels(data_class$alpha_index)

pdf("alpha_boxplot_class.pdf", useDingbats=FALSE, onefile=T,
paper='A4r', width=12, height=3)
ggplot(data_class, aes(x = class, y = value)) +
geom_boxplot(aes(fill=class, colour=class), alpha=0.4,outlier.colour =
"red", outlier.shape = 8) + facet_wrap(~alpha_index, scales="free")+
scale_fill_manual(values = c("#00AFBB", "#E7B800", "#FC4E07",
"magenta4", "seagreen")) +scale_colour_manual(values= c("#00AFBB",
"#E7B800", "#FC4E07", "magenta4", "seagreen"))+
theme_light()+theme(legend.position="none")  + theme(strip.background =
element_rect(fill="grey85"))+ theme(strip.text.x = element_text(colour
= "black", face = "bold"), strip.text.y = element_text(colour =
"black", face = "bold")) +labs(x = "Sample Type",y = "",color = "")
+geom_text(data=data_class, aes(x=class, y=response, label=.group), vjust=-
2.4, hjust=1, check_overlap = T, show.legend = FALSE, size=3.5)

dev.off()

#time
data_time=merge(data, cld_dtime, by=c("alpha_index","time"))
selected=c("sobs","shannoneven")
data_time1= data_time[data_time$alpha_index%in%selected,]
data_time1$alpha_index =droplevels(data_time1$alpha_index)

selected=c("invsimpson")
data_time2= data_time[data_time$alpha_index%in%selected,]
data_time2$alpha_index =droplevels(data_time2$alpha_index)
data_time2$.group=c("")

data_time_final=rbind(data_time1, data_time2)

pdf("alpha_boxplot_time.pdf", useDingbats=FALSE, onefile=T,
paper='A4r', width=12, height=3)
ggplot(data_time_final, aes(x = time, y = value)) +
geom_boxplot(aes(fill=time, colour=time), alpha=0.4,outlier.colour =
"red", outlier.shape = 8) + facet_wrap(~alpha_index, scales="free")+
scale_fill_manual(values = c("#00AFBB", "#E7B800", "#FC4E07",
"magenta4", "seagreen","purple4")) +scale_colour_manual(values=
c("#00AFBB", "#E7B800", "#FC4E07", "magenta4", "seagreen","purple4"))+
theme_light()+theme(legend.position="none")  + theme(strip.background =
```

```
element_rect(fill="grey85"))+ theme(strip.text.x = element_text(colour
= "black", face = "bold"), strip.text.y = element_text(colour =
"black", face = "bold")) +labs(x = "Sample Type",y = "",color = "")
+geom_text(data=data_time_final, aes(x=time, y=response, label=.group), vjust=-
2.7, hjust=1, check_overlap = T, show.legend = FALSE, size=3.5)

dev.off()
```

#Rarefaction (FIG S1): assessing the depth of coverage for each sample grouped according to sample type and sampling time.

```
rm(list=ls())
library(tidyverse)
library(RColorBrewer)
library(data.table)
library(ggforce)

#read
rare=read.table(file="RumenTimecourse_Norm.groups.rarefaction",
header=TRUE)
meta_data=read.table(file="meta_data_Joe.txt", header=TRUE, sep = "\t")

#subset
rownames(rare)=rare$numsampled
subset=rare %>% select(contains("X0.03."))
colnames(subset)=gsub("X0.03.","", colnames(subset))
subset$numsampled=rownames(subset)
subset$numsampled=as.numeric(subset$numsampled)
rownames(subset)=NULL

#merge/melt
melt=melt(subset, id.vars="numsampled")
colnames(melt)=c("numsampled", "group","sobs")
melt=merge(meta_data, melt, by="group")
melt <- na.omit(melt)
melt$class=as.factor(melt$class)
melt$time=as.factor(melt$time)


#Plot
color= colorRampPalette(
c("#00AFBB","#E7B800","#FC4E07","purple","seagreen"))(5)
pdf("code/rarefaction_by_class_and_time.pdf", useDingbats=FALSE,
onefile=T, paper='A4', width=9, height=11)
ggplot(melt, aes(x = numsampled, y = sobs))+
  geom_line(aes(x = numsampled, y = sobs, group = group, color =
class), size=1,linetype=1,alpha=0.4)+
  geom_text(data = melt %>%
              group_by(group, class, time) %>%
```

```
              summarise(max_OTU = max(sobs),
              max_raw = max(numsampled)),
              aes(x = max_raw, y = max_OTU, label = group),
check_overlap = T, hjust = 0, size=1)+  theme_bw()+facet_wrap (~ time,
ncol=2) + labs(x = "Number of Sequences",y = "Number of OTUs",colour =
"Sample type")  + scale_colour_manual(values=color) +
theme_classic()+theme(legend.position="right", strip.background =
element_rect(fill="grey85"))
dev.off()
```

#Beta diversity-PCoA (Fig. 1): visualizing differences in the Bray-Curtis compositional dissimilarity of bacterial communities according to sample type and sampling time

```
rm(list=ls())
library(ape)
library(vegan)
library(EcolUtils)
library(ggplot2)
library(RColorBrewer)

#read OTU table and meta data
otu=read.table(file="RumenTimecourse_Norm.shared_clean.txt",
header=TRUE)
meta_data=read.table(file="meta_data_Joe.txt", header=TRUE, sep = "\t")
rownames(otu)=otu$Group; otu$Group=otu$label=otu$numOtus=NULL
meta_data= meta_data[match(rownames(otu),rownames(meta_data)),]

#PCoA
dist <- vegdist(otu, "bray")
pcoa <- pcoa(dist, correction="none")
vector=as.data.frame(pcoa$vectors[,1:2])
PCOA=data.frame(group=rownames(vector), x=vector
[,1],y=vector[,2],class=as.factor(meta_data[,3]),
time=as.factor(meta_data[,4]))

PCoA1_eig=round(100*(pcoa$values$Relative_eig[1]), 2)
PCoA2_eig=round(100*(pcoa$values$Relative_eig[2]), 2)

x_lab=paste("PCoA1", "(", PCoA1_eig, "%",")",sep="")
y_lab=paste("PCoA2", "(", PCoA2_eig, "%",")",sep="")

#Plot
pdf(file="code/PCOA_class_time.pdf", onefile=T, paper='A4', width=8.27,
height=11.69)
ggplot(PCOA, aes(x = x, y = y, color = class, fill=class)) +
geom_point(aes(color=factor(class), shape=factor(class),
size=factor(class)), alpha=0.6) + geom_point(colour= c("white"),
size=0.015, stat="identity") + scale_color_manual(values=c("#00AFBB",
```

```
"#E7B800", "#FC4E07","#E7B800", "#FC4E07")) +
scale_size_manual(values=c(3.5,2.5,2.5,3,3))
+scale_shape_manual(values=c(18,16,16, 17,17),guide="legend")
+theme(plot.background=element_rect(fill="white"),plot.margin =
unit(c(0,0,0,0), "cm"))+ theme(legend.position="right") +  labs(x=
x_lab ,y = y_lab, colour ="Sample type", shape="Sample type",
fill="Sample type") + scale_y_continuous(limits = c(-
0.9,0.9),breaks=seq(-0.9,0.9, by=0.5))+scale_x_continuous(limits = c(-
0.9,0.9), breaks=seq(-0.9, 0.9, by=0.5)) +facet_wrap( ~ time, nrow =3)
+ guides(colour = "legend",  shape = "legend")+ guides(color =
guide_legend("Sample type"), shape = guide_legend("Sample type"), size=
guide_legend("Sample type"))+theme_classic()+ theme(strip.background =
element_rect(fill="grey85")) + coord_fixed()+ stat_ellipse(type =
"norm", linetype ="dashed", alpha=0.5, size=0.25)
dev.off()
```

#continuing Beta diversity-Permanova (TABLE S3): Assessing differences in the Bray-Curtis compositional dissimilarity of bacterial communities in response to sampling time, sample type and interaction of these two factors

```
meta_data$cow=as.factor(meta_data$cow)
meta_data$time=as.factor(meta_data$time)
meta_data$class=as.factor(meta_data$class)

permanova=adonis2(dist ~ meta_data$time*meta_data$class,
permutations=1000, method="bray", by = "terms")
saveRDS(permanova, file="code/permanova_model")

ad=adonis.pair(dist, meta_data$time*meta_data$class, nper = 1000,
corr.method = "BH")

library(splitstackshape)
ad=cSplit(ad, "combination", "<->")
ad=as.data.frame(ad)
ad$combination_2=NULL

library(dplyr)
pairwise=rbind(filter(ad, grepl('T1',combination_1) &
grepl('T1',combination_3)), filter(ad, grepl('T2',combination_1) &
grepl('T2',combination_3)), filter(ad, grepl('T3',combination_1) &
grepl('T3',combination_3)), filter(ad, grepl('T4',combination_1) &
grepl('T4',combination_3)), filter(ad, grepl('T5',combination_1) &
grepl('T5',combination_3)), filter(ad, grepl('T6',combination_1) &
grepl('T6',combination_3)))

#export
library(openxlsx)
write.xlsx(pairwise, file = "code/permanova_pairwise.xlsx")
```

```
# Random Forest classifier (Excel TABLE S4, S5, S6): 3 merged classes
rm(list=ls())
library(DMwR)
library(data.table)
library(caret)


#Read
abund=read.table(file="abund_filtered_0.05_0_norm.txt", header=TRUE,
sep = "\t")
meta_data=read.table(file="meta_data_Joe.txt", header=TRUE, sep = "\t")
meta_data= meta_data[match(rownames(abund),rownames(meta_data)),]

#Relevel
levels(meta$class) <- c("BS","RL","RS","RL","RS")
table(meta_data$class)

#Add class to dataset
abund$class=meta_data[rownames(abund), "class"]

# Split into Train and Validation sets (70 : 30 random)
intrain <- createDataPartition(abund$class,p=0.7, list=F)
train_data <- abund[intrain,]
test_data <- abund[-intrain,]
train_class <- as.factor(train_data$class)
test_class <- as.factor(test_data$class)
train_data <- train_data[,-which(colnames(train_data)=="class")]
test_data <- test_data[,-which(colnames(test_data)=="class")]

set.seed(100)

# replace (sampling="down","up","smote"), run a save model and
confusion matrix in testing set

#Model default
model_default <- caret::train(train_data, train_class, method = "rf",
trControl = trainControl(method = "repeatedcv", repeats = 3, number =
10, verboseIter = T, sampling = "none", classProbs = TRUE,
savePredictions = TRUE))
prob_default <- data.frame(observed = test_class,
predict(model_default, test_data, type = "prob"))
prob_default$predicted= predict(model_default,test_data)
prob_default$match <- prob_default$observed ==prob_default$predicted
cm_default <- confusionMatrix(prob_default$predicted, test_class)
saveRDS(model_default, "model_default_final.txt")
saveRDS(cm_default, "cm_default_final.txt")

#Visualize overall performance (Kappa and Accuracy)
model_smote =readRDS("model_smote_final.txt")
model_up=readRDS("model_up_final.txt")
model_default=readRDS("model_default_final.txt")
model_down=readRDS("model_down_final.txt")
```

```
models <- list(default = model_default, up = model_up, down =
model_down, smote = model_smote)
resampling <- resamples(models)
bwplot(resampling)
dev.off()

# Visualize model performance by sample type
#replace default per smote, down and up
default<- data.frame(Sensitivity = cm_default$byClass[,"Sensitivity"],
           Specificity = cm_default$byClass[,"Specificity"],
           Precision = cm_default$byClass[,"Precision"],
           Recall = cm_default$byClass[,"Recall"],
           F1 = cm_default$byClass[,"Balanced Accuracy"])
default$model=c("default")

plot=rbind(default, smote, up, down)
plot$class=rownames(plot);rownames(plot)=NULL

library(dplyr)
library(ggplot2)
library(tidyr)

plot1=plot %>% gather(x, y, Sensitivity:Balanced Accuracy)
plot1$class=as.factor(plot1$class)
plot1$model=as.factor(plot1$model)

library(openxlsx)
write.xlsx(plot1, file = "code/Random_Forest_results.xlsx")


#VIMP plot
library (dplyr)
imp=as.data.frame(varImp(model_smote)$importance, varImp.train=FALSE)
imp$OTU = as.factor(rownames(imp));rownames(imp)=NULL
arranje <- arrange(imp, desc(Overall))
top20 <- arranje [1:24,]#select OTUs that display importance>=50%

pdf(file="vimp_plot.pdf", onefile=T, paper='A4r', width=11, height=5)
color2= colorRampPalette( c("#00AFBB", "#E7B800","#FC4E07"))(24)

ggplot(df)+ geom_linerange(aes(x= reorder(taxon, Overall), ymin=0,
ymax=Overall ,colour=taxon), size=1.0, alpha=0.5)+
    geom_point(aes(x =taxa, y = Overall, colour =taxa), size=3,
alpha=0.5, shape=21, fill="white", stroke=1.5) +
theme_light()+scale_colour_manual(values=color2) + labs(x = "Taxa",
y="% MeanDecreaseGini")+ theme(strip.background =
element_rect(fill="grey85"))+coord_flip()+
theme(legend.position="none") +theme(text = element_text(size=12))
```

```
dev.off()
```

 # ANOVA (Excel TABLE S7): Assessing differences in the relative abundance of bacterial OTUs using General Linear Mixed Model estimated via Penalized Quasi-Likelihood. Differences in alpha diversity were assessed using the same model under Gamma distribution (TABLE S2) instead Poisson. In both analyses, the covariance structure used was corAR1: autocorrelation structure of order 1 that requires equally spaced time intervals.

```
rm(list=ls())
library(MASS)
library(multcomp)
library(nlme)
library(lsmeans)
library(ggplot2)
```

#read

```
cont=read.table(file="otus_filtered_0.05_0.txt", header=TRUE, sep = "\t")
meta=read.table(file="meta_data_Joe.txt", header=TRUE, sep = "\t")
meta=meta[match(rownames(cont),rownames(meta)),]
taxa= read.table(file="taxa_fixed_novo.txt", header=TRUE, sep = "\t")
taxa=subset(taxa, select=c("OTU","novo"))
```

#offset variable

```
ultimo=length(cont)
tot_bac <- apply(cont[,1:length(cont)],1,sum)
```

#Sub setting OTUs present in at least 80% of all samples

```
shared=colSums(cont!= 0)
shared_otus= names(shared[shared>= 185]) samples
cont<- cont[,shared_otus];
dim(cont)
```

#Anova with post hoc (Tukey correction)

```
dat=cbind(meta, cont)
dat$cow=as.factor(dat$cow)
dat$class=as.factor(dat$class)
dat$time=as.factor(dat$time)

set.seed(3)
resp_bac_cont <- names(dat)[!(names(dat) %in% c("cow","time", "class",
"group"))]

analise<- lsm_dtime <- lsm_dclass <- lsm_dint <- cld_dtime <-cld_dclass <-
cld_dint<- contr_dtime <- contr_dclass <- contr_dint <-NULL

for(i in resp_bac_cont){
bac<-data.frame(dat[2:4],y=dat[,i])
fit<-glmmPQL (y~time*class+offset(log(tot_bac)), random=~1|cow/time,
family=poisson(link = log), correlation=corAR1(), data = bac)
anova=anova.lme(fit, type="sequential", adjustSigma=FALSE)
analise<-rbind(analise,data.frame("OTU"=i,anova))
```

```
lsm_time <- lsmeans(fit, ~ time, adjust="tukey")
lsm_dtime<-rbind(lsm_dtime,data.frame("OTU"=i, lsm_time))
contr_time=contrast(lsm_time, alpha=0.05, method="pairwise")
contr_dtime<-rbind(contr_dtime,data.frame("OTU"=i, contr_time))
cld_time <- cld(lsm_time, Letters=letters, reverse=T, alpha=0.05)
cld_dtime<- rbind(cld_dtime,data.frame("OTU"=i, cld_time))

lsm_class <- lsmeans(fit, ~ class, adjust="tukey")
lsm_dclass<-rbind(lsm_dclass,data.frame("OTU"=i, lsm_class))
contr_class=contrast(lsm_class, alpha=0.05, method="pairwise")
contr_dclass<-rbind(contr_dclass,data.frame("OTU"=i, contr_class))
cld_class <- cld(lsm_class, Letters=letters, reverse=T, alpha=0.05)
cld_dclass<- rbind(cld_dclass,data.frame("OTU"=i, cld_class))

lsm_int <- lsmeans(fit, ~ class|time, adjust="tukey")
lsm_dint<-rbind(lsm_dint,data.frame("OTU"=i, lsm_int))
contr_int=contrast(lsm_int, alpha=0.05, method="pairwise")
contr_dint<-rbind(contr_dint,data.frame("OTU"=i, contr_int))
cld_int <- cld(lsm_int, Letters=letters, reverse=T, alpha=0.05)
cld_dint<- rbind(cld_dint,data.frame("OTU"=i, cld_int))
}

#Adjusting p-values
analise$factor=c("(Intercept)","class","time","class:time")
rownames(analise)=NULL
analise=subset(analise, select=c(1, 6, 2:5))
analise=subset(analise, factor!= "(Intercept)")
analise$p.adjust= p.adjust(analise$p.value, "fdr")
analise=subset(analise, p.adjust<=0.05)

#Sub setting final results
anova=merge(taxa, analise, by="OTU")
contr_dint= merge(taxa, contr_dint, by="OTU")
contr_dtime= merge(taxa, contr_dtime, by="OTU")
contr_dclass= merge(taxa, contr_dclass, by="OTU")

anova_time=subset(anova, factor=="time")
anova_class=subset(anova, factor=="class")
anova_int=subset(anova, factor=="class:time")

names_int_time=dput(as.character(intersect(unique(anova_int$OTU),
unique(anova_time$OTU))))

names_int_class=dput(as.character(intersect(unique(anova_int$OTU),
unique(anova_class$OTU))))


#varied independently of interaction
anova_class1= anova_class[!anova_class$OTU%in%names_int_class,]
anova_time1 = anova_time[!anova_time$OTU%in%names_int_time,]
```

```
names_anova_int=dput(as.character(anova_int$OTU))
names_anova_time=dput(as.character(anova_time1$OTU))
names_anova_class=dput(as.character(anova_class1$OTU))

cld_dint2= cld_dint[cld_dint$OTU%in%names_anova_int,]
contr_dint2= contr_dint[contr_dint$OTU%in%names_anova_int,]
lsm_dint2= lsm_dint[lsm_dint$OTU%in%names_anova_int,]
cld_dclass2= cld_dclass[cld_dclass$OTU%in%names_anova_class,]
contr_dclass2= contr_dclass[contr_dclass$OTU%in%names_anova_class,]
lsm_dclass2= lsm_dclass[lsm_dclass$OTU%in%names_anova_class,]
dput(as.character(unique(contr_dclass2$OTU)))

cld_dtime2= cld_dtime[cld_dtime$OTU%in%names_anova_time,]
contr_dtime2= contr_dtime[contr_dtime$OTU%in%names_anova_time,]
lsm_dtime2= lsm_dtime[lsm_dtime$OTU%in%names_anova_time,]
dput(as.character(unique(contr_dtime2$OTU)))

#Combining final result to export
anova1=rbind(anova_int, anova_time1, anova_class1)

#Export
library(openxlsx)
results <- list(
"anova" = anova1,
"lsmeans_time"=lsm_dtime2, "letters_time" = cld_dtime2, "contrast_time"=
contr_dtime2,
"lsmeans_class"=lsm_dclass2,"letters_class"= cld_dclass2,"contrast_class"=
contr_dclass2,
"lsmeans_int"=lsm_dint2,"letters_int"= cld_dint2, "contrast_int"=
contr_dint2)

write.xlsx(results, file = "code/Anova_POISSON_OFFSET_Results.xlsx")
```