Temporal and Spatial Distribution of Harbor Seal (*Phoca vitulina*) Risk of Entanglement in Gillnets Along the Norwegian Coast

Supplemental Appendices

Jonas O. Elnes,[1, 2] André Moan,[1, 2] Kjell T. Nilssen,[1]
L. Asbjørn Vøllestad,[2] and Arne Bjørge[1]

[1]*Department of Marine Mammals, Institute of Marine Research, PO Box 1066, Blindern, NO-0316 Oslo, Norway*
E-mail: arne.bjoerge@hi.no
[2]*Department of Biosciences, University of Oslo, PO Box 1066, Blindern, NO-0316 Oslo, Norway*

Supplemental Appendix I

Information on Statistical Sea Locations with High Interaction Risks (Table A.1) and Data on Harbor Seals from Each Molt Site (Table A.2)

Table A.1. An overview of Statistical Sea Locations (SSLs) predicted with high interaction risk (RC6 and RC7). A total of 14 SSLs were predicted with high interaction risk in a minimum of one season. A total of 25 combinations of season and SSL were predicted with high interaction risk. Each SSL is divided in the season predicted with high bycatch risk, where W = winter, Sp = spring, Su = summer, and A = autumn. Hs represents the relative vulnerable abundance of harbor seals (*Phoca vitulina*) in the cell. Hauls are the average abundance of trips in the cell. The interaction values are derived from Equation 3 (see article) and explain the proportion of risk explained by the season and SSL combination when all season and SSL combinations equal one.

SSL	Season	Hs	Hauls	Interaction value
00-05	Su	246	110	0.018
	A	115	149	0.020
03-05	Sp	55	564	0.036
03-24	Sp	28	377	0.012
05-24	W	31	524	0.020
	Su	116	242	0.020
	A	55	188	0.012
05-25	W	33	301	0.012
	Su	108	99	0.013
	A	56	353	0.023
05-30	W	8	1,295	0.012
05-31	W	119	381	0.011
06-18	Su	50	175	0.010
	A	24	407	0.011
07-07	Su	38	443	0.020
07-08	Su	110	103	0.013
07-25	W	67	256	0.020
	Sp	48	280	0.016
	Su	238	236	0.066
	A	114	300	0.040
09-16	W	30	308	0.011
	Sp	21	542	0.013
	Su	93	385	0.042
09-20	A	163	102	0.020
28-03	A	195	107	0.024

Table A.2. Harbor seal count data from the Institute of Marine Research

County	Municipality	Area	Locality	Latitude	Longitude	Total	Date (d/mo/y)
Finnmark	Hasvik	Meltefjord		70.5049	22.4006	12	13/8/2021
Finnmark	Hasvik	Øyfjord		70.5381	22.6711	86	13/8/2021
Finnmark	Hasvik Ø	Husfjord		70.5208	22.6711	2	13/8/2021
Finnmark	Hasvik	Galtefjorden		70.6914	22.6431	56	17/8/2021
Finnmark	HasvikV	Bølefjorden		70.7193	22.5153	25	17/8/2021
Finnmark	Hammerfest	Lille Kamøy		70.8518	23.06	34	17/8/2021
Finnmark	Hammerfest	Lille Kamøy		70.8478	23.0708	51	17/8/2021
Finnmark	Hammerfest	Store Kamøy		70.846	23.0136	26	17/8/2021
Finnmark	Hammerfest	Store Kamøy		70.8306	23.0526	4	17/8/2021
Finnmark	Måsøy	Kobbeøya		70.9176	25.2784	9	18/8/2021
Finnmark	Måsøy	Kobbeøya		70.918	25.3067	127	18/8/2021
Finnmark	Måsøy	Kobbefjorden		70.9347	25.3357	3	18/8/2021
Finnmark	Porsanger	Valdak		70.1453	24.9175	81	23/8/2021
Finnmark	Porsanger	Langbukta		70.0655	25.1508	0	23/8/2021
Finnmark	Porsanger	Cohkaskarku		70.1629	25.1784	0	23/8/2021
Finnmark	Porsanger	Store Ørnøy		70.1888	25.299	112	23/8/2021
Finnmark	Porsanger	Skogholmen		70.2984	25.4665	22	23/8/2021
Finnmark	Lebesby	Bekkarfjord		70.6197	27.1402	88	24/8/2021
Finnmark	Lebesby	Ifjorden		70.4875	26.9148	0	24/8/2021
Finnmark	Berlevåg	Kongsfjord		70.6767	29.2729	122	26/8/2021
Finnmark	Tana	Høgholmen		70.5231	28.4335	228	27/8/2021
Finnmark	Tana	Høgholmen		70.5231	28.4335	230	27/8/2021
Møre og Romsdal	Sandøy	Ona - Sandøya	Hareidsholmene	62.8361	6.6109	32	14/8/2018
Møre og Romsdal	Sandøy	Ona - Sandøya	Hareidsholmene	62.8343	6.6178	20	14/8/2018
Møre og Romsdal	Sandøy	Ona - Sandøya	Hummarskjæra	62.8492	6.5686	6	14/8/2018
Møre og Romsdal	Sandøy	Harøya nordvest	Erkneskjæret	62.7925	6.3433	0	14/8/2018
Møre og Romsdal	Sandøy	Bollana sør	Knappane	62.79687	6.4045	5	14/8/2018
Møre og Romsdal	Sandøy	Harøya vest	Raukollene	62.78038	6.4003	9	14/8/2018
Møre og Romsdal	Sandøy	Bollane vest	Leiskjærene	62.80538	6.3861	19	14/8/2018
Møre og Romsdal	Sandøy	Harøya nordvest	Erkneskjæra	62.79	6.3433	13	14/8/2018
Møre og Romsdal	Sandøy	Harøya vest	Havsteinen	62.7874	6.2837	48	14/8/2018
Møre og Romsdal	Haram	Nordøyane	Brunkleggen	62.7385	6.3219	16	14/8/2018
Møre og Romsdal	Haram	Nordøyane	Kobbskjæran	62.73514	6.3083	14	14/8/2018
Møre og Romsdal	Haram	Nordøyane	Aglapen	62.74401	6.2936	53	14/8/2018
Møre og Romsdal	Haram	Nordøyane	Svetlingane	62.7174	6.2407	56	14/8/2018
Møre og Romsdal	Haram	Nordøyane	Løvøyklovningen	62.6378	6.0361	41	14/8/2018
Møre og Romsdal	Sande	Kvamsøy	Eggholmen Notaskjær	62.19917	5.36072	40	27/8/2018
Møre og Romsdal	Ulstein	Grasøyene	Tøyrane	62.4197	5.7379	26	27/8/2018
Møre og Romsdal	Giske	Erkna - Vigra	Rossholmen Lille	62.6027	6.0512	34	27/8/2018
Møre og Romsdal	Aukra	Saltsteinen	Synnøvskjær	62.8714	6.6621	10	28/8/2018
Møre og Romsdal	Aukra	Saltsteinen	Sandskjær	62.87621	6.6809	50	28/8/2018
Møre og Romsdal	Aukra	Saltsteinen	Saltsteinkråka	62.8932	6.6711	24	28/8/2018
Møre og Romsdal	Averøy	Orskjæra	Fonnskaftet. Littlvasshl	63.1069	7.2419	22	28/8/2018
Møre og Romsdal	Averøy	Orskjæra	Maltskjæra/ Brattskjerdingen	63.1052	7.2796	48	28/8/2018

Møre og Romsdal	Averøy	Orskjæra	Buholmen	63.1032	7.2759	26	28/8/2018
Nord - Trøndelag	Leka	Hortavær	Ertenbalen	65.2796887	11.35959	19	15/8/2019
Nord - Trøndelag	Leka	Hortavær	Sølbuflesa/ Fuholmen	65.2294566	11.3657191	20	15/8/2019
Nord - Trøndelag	Leka	Hortavær	Lissrundholmen	65.2245958	11.3945709	14	15/8/2019
Nordland	Gildeskål	Sør Fugløy	Floholmen skj mellom Ytre-Indre F	67.0208177	13.6580091	38	12/8/2019
Nordland	Gildeskål	Sør Fugløy	Skj mellom Måøy-Langskj-Trollskj	67.0515665	13.5826483	150	12/8/2019
Nordland	Gildeskål	Sør Fugløy	Nattmålskj-Brokholmskj	67.0691424	13.6590637	241	12/8/2019
Nordland	Gildeskål	Sør Fugløy	Floholmen skj mellom Ytre-Indre F	67.0208177	13.6580091	13	13/8/2019
Nordland	Gildeskål	Sør Fugløy	Skj mellom Måøy-Langskj-Trollskj	67.0515665	13.5826483	152	13/8/2019
Nordland	Gildeskål	Sør Fugløy	Nattmålskj-Brokholmskj	67.0691424	13.6590637	88	13/8/2019
Nordland	Rødøy	Nesøy sør	Skj NV for Langskjæret	66.5556076	12.5346004	58	13/8/2019
Nordland	Dønna	Vandve sør	Kuflesan	66.1724847	12.2111114	19	14/8/2019
Nordland	Dønna	Vandve sør	Kuflesan	66.1724847	12.2111114	15	14/8/2019
Nordland	Brønnøy	Onstein	Skj ved Stokkskjær	65.4227473	11.7488415	66	16/8/2019
Nordland	Vega	Fugløyvær	Fugleværet - sør - nord	65.6333	11.5693	2	16/8/2019
Nordland	Brønnøy	Henstein	Askjellskjær	65.3858	11.7001	8	17/8/2019
Nordland	Lurøy	Nesøy	Skjærflesa	66.4839	12.5639	13	17/8/2020
Nordland	Lurøy	Nesøy	Ertenflesa	66.5177	12.5974	32	17/8/2020
Nordland	Lurøy	Nesøy	Skjær mellom Lyngøy og Kloholmen	66.5393	12.5966	17	17/8/2020
Nordland	Røst	Røst	Småskjæran	67.4955	11.9791	13	20/8/2020
Nordland	Røst	Røst	Inner Låtra	67.4846	11.9135	112	20/8/2020
Nordland	Røst	Røst	Ytter Låtra	67.4856	11.8887	19	20/8/2020
Nordland	Røst	Røst	Eflatskjæran	67.431	11.8607	16	20/8/2020
Nordland	Røst	Røst	Kobbholmen. Låtra	67.5219	12.1895	15	20/8/2020
Nordland	Hadsel	Ongstadvika	Klemningan	68.4928	14.6804	52	22/8/2020
Nordland	Hadsel	Ongstadvika	Talgskjærene	68.5272	14.6568	36	22/8/2020
Nordland	Andøy	Andøya øst	Myrflesa	68.0949	16.0643	1	24/8/2020
Nordland	Andøy	Risøyrenna	Åseflesa	68.9918	15.8038	93	24/8/2020
Nordland	Sortland	Kvæfjord	Kinnholmen	68.9369	15.9899	24	24/8/2020
Nordland	Andøy	Andøya Vest	Bardan	69.3361	16.0589	0	25/8/2020
Nordland	Andøy	Andøya Vest	Moholmen	69.3259	16.0632	0	25/8/2020
Nordland	Andøy	Skogvoll	Staveøyan	69.1869	15.7784	9	25/8/2020
Nordland	Andøy	Skogvoll	Klonkholmen	69.1657	15.7695	299	25/8/2020
Nordland	Andøy	Skogvoll	Nesholman	69.1741	15.8024	118	25/8/2020
Nordland	Andøy	Skogvoll	Skarvklakken	69.1582	15.6735	0	25/8/2020
Sogn og Fjordane	Nordfjord	Utvikfjorden	Nosanaustet	61.84156	6.37872	13	17/8/2018
Sogn og Fjordane	Nordfjord	Utvikfjorden	Tosken	61.84964	6.3631	9	17/8/2018
Sogn og Fjordane	Nordfjord	Utvikfjorden	Tosken	61.84964	6.3631	28	17/8/2018
Sogn og Fjordane	Nordfjord	Utvikfjorden	Tosken	61.84964	6.3631	8	17/8/2018
Sogn og Fjordane	Flora	Hovden øst	Gåsøyskjæra	61.6755	4.9375	26	19/8/2018
Sogn og Fjordane	Flora	Hovden øst	Grunnane	61.6809	4.9021	2	19/8/2018

Sogn og Fjordane	Sognefjorden	Aurlandsfjorden	Flåm	60.87	7.1264	2	23/8/2018
Sogn og Fjordane	Sognefjorden	Aurlandsfjorden	Østsida ytterst	61.0061	7.0693	5	23/8/2018
Sogn og Fjordane	Sognefjorden	Sognefj. Sørsida		61.118	7.1972	13	23/8/2018
Sogn og Fjordane	Sognefjorden	Sognefj.nord v. fergeleiet		61.1741	7.3862	8	23/8/2018
Sogn og Fjordane	Sognefjorden	Lusterfjorden	Svartheller	61.1951	7.3566	2	23/8/2018
Sogn og Fjordane	Solund	Nessefjorden	n Helleholmene V	61.0855	4.9082	4	23/8/2018
Sogn og Fjordane	Solund	Ospa	Ospøytåa	61.2225	4.719	10	23/8/2018
Sogn og Fjordane	Askvoll	Smellvær	Kråka/Godskjæret	61.42559	4.92391	7	24/8/2018
Sogn og Fjordane	Askvoll	Smellvær	Ognholmen	61.40268	4.88695	60	24/8/2018
Sogn og Fjordane	Askvoll	Moldvær	Melskjær	61.38775	4.73902	34	24/8/2018
Sogn og Fjordane	Askvoll	Ryggsteinen	Stemmeskjær	61.3986	4.6639	47	24/8/2018
Sogn og Fjordane	Askvoll	Ryggsteinen	Ytreøyna	61.3958	4.68943	17	24/8/2018
Sogn og Fjordane	Askvoll	Håstein	Skarveskjær	61.351	4.6573	127	24/8/2018
Sogn og Fjordane	Askvoll	Kvitingane	Nordre Hjelparen	61.29593	4.8708	13	24/8/2018
Sogn og Fjordane	Askvoll	Kvitingane	Søndre Hjelparen	61.29154	4.8777	25	24/8/2018
Sogn og Fjordane	Askvoll	Kvitingane	Vardholmen	61.2871	4.8635	38	24/8/2018
Sogn og Fjordane	Askvoll	Kvitingane	Vardhl N/ Grønneskjæret	61.2871	4.8635	107	24/8/2018
Sogn og Fjordane	Askvoll	Atløyna	Skardvikholmen	61.3337	4.8875	1	24/8/2018
Sogn og Fjordane	Askvoll	Værøyna	Harbakken/N Sædingskjær	61.2772	4.7082	11	24/8/2018
Sogn og Fjordane	Flora	Hovden øst	Gåsøyskjæra	61.6755	4.9375	19	26/8/2018
Sør - Trøndelag	Frøya	Flesan		63.9577	8.9739	41	19/8/2019
Sør - Trøndelag	Frøya	Gjesingen		63.9397	8.9246	11	19/8/2019
Sør - Trøndelag	Frøya	Gjesingen		63.9397	8.9246	4	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9535	8.783	21	19/8/2019
Sør - Trøndelag	Frøya	Gronga	Beinhl/Midt-Gronga	63.9535	8.783	25	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9535	8.783	17	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9535	8.783	13	19/8/2019
Sør - Trøndelag	Frøya	Gronga	Tromsa	63.9164	8.8554	14	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9164	8.8554	10	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9164	8.8554	10	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9164	8.8554	16	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9164	8.8554	25	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9164	8.8554	3	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9164	8.8554	30	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9164	8.8554	18	19/8/2019
Sør - Trøndelag	Frøya	Gronga		63.9164	8.8554	3	19/8/2019
Sør - Trøndelag	Frøya	Gjesingen		63.9535	8.783	23	19/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	15	20/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	6	20/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	10	20/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	23	20/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	13	20/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	16	20/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	2	20/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	15	20/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	7	20/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	10	20/8/2019

Sør - Trøndelag	Frøya	Sørburøy i Froan		63.9776	8.9926	24	20/8/2019
Sør - Trøndelag	Frøya	Sørburøy i Froan	Sauøya	63.7518	8.6399	27	20/8/2019
Sør - Trøndelag	Bjugn	Tarva	Undtarva	63.8072	9.5231	343	20/8/2019
Sør - Trøndelag	Bjugn	Tarva	Tarva øst	63.8137	9.4959	141	20/8/2019
Sør - Trøndelag	Namsos	Innergåsa	Styrmannskjær	64.466	11.3768	1	16/8/2020
Sør - Trøndelag	Namsos	Innergåsa	Brannøya	64.4552	11.3908	3	16/8/2020
Sør - Trøndelag	Namsos	Straumsneset	Dragøy	64.4138	11.3537	36	16/8/2020
Sør - Trøndelag	Namsos	Kjølsøya	Kobbskjær	64.3985	11.3779	1	16/8/2020
Sør - Trøndelag	Namsos	Røyklibotn	Storskjær	64.607	11.6126	21	16/8/2020
Sør - Trøndelag	Namsos	Røyklibotn	Tennholmen	64.6146	11.6052	9	16/8/2020
Troms	Kvæfjord	Kvæfjord	Kasholmen	68.8879	16.172	37	24/8/2020
Troms	Tromsø	Lyngøya		69.9297	18.5061	57	21/8/2020
Troms	Tromsø	Risøya/Sandvær		69.9577	18.5405	207	21/8/2020
Troms	Karlsøy	Grimsholmen		70.2877	19.7788	236	26/8/2020
Troms	Karlsøy	Måsvær		70.218	18.9938	89	26/8/2020
Troms	Senja	Holmenvær		69.2641	16.7761	50	29/8/2020
Troms	Senja	Bergsøyene		69.4847	17.1866	42	30/8/2020
Troms	Senja	Kobbeøyra/Gisundet		69.326	17.9828	5	31/8/2020
Troms	Senja	Hofsøybotn		69.0499	17.0956	4	6/9/2020
Ø-Agder		Sørfjord	Styrtholmen	58.735	9.14055556	14	19/8/2016
Ø-Agder		Askerøy V	Selskjær	58.6169444	9.07388889	14	19/8/2016
Ø-Agder		Askerøy Ø	Hengsholmen	58.6108333	9.08138889	10	19/8/2016
Østfold		Hvalerområdet	Hvalerområdet	59.069537	10.966219	337	19/8/2016
Rogaland		Tysvær	Årvikholmen. Kårstø	59.2588889	5.5344	22	18/8/2017
Rogaland		Kvitsøy	Gåsholmen	59.0425	5.46888889	65	18/8/2017
Rogaland		Kvitsøy	Rauholmen	59.0130556	5.42	61	18/8/2017
Rogaland		Kvitsøy	Magerøya	59.0144444	5.39111111	5	18/8/2017
Rogaland		Kvitsøy	Vestre Imsen	59.0091667	5.37	38	18/8/2017
Rogaland		Sola	Store Sandøy	58.9561111	5.46888889	34	18/8/2017
Rogaland		Sola	Håstein skjær N	58.9527778	5.44805556	63	18/8/2017
Rogaland		Frafjord	Østside	58.84778	6.2155	1	21/8/2017
Rogaland		Bokn	Kårstø. Store Rova	59.2283333	5.5408	15	21/8/2017
Rogaland		Lysefjorden	Nordside	58.9997222	6.22444444	7	28/8/2017
Rogaland		Lysefjorden	Nordside	59.0166667	6.24388889	1	28/8/2017
Rogaland		Lysefjorden	Nordside	59.045	6.46222222	1	28/8/2017
Rogaland		Lysefjorden	Nordside	59.0333333	6.47388889	8	28/8/2017
Rogaland		Lysefjorden	Nordside	59.0369444	6.48944444	5	28/8/2017
Rogaland		Lysefjorden	Nordside	59.0388889	6.50583333	2	28/8/2017
Rogaland		Lysefjorden	Nordside	59.0444444	6.54666667	12	28/8/2017
Rogaland		Lysefjorden	Nordside	59.0461111	6.57611111	7	28/8/2017
Rogaland		Lysefjorden	Nordside	59.0461111	6.55555556	6	28/8/2017
Rogaland		Lysefjorden	Nordside	59.0461111	6.55277778	2	28/8/2017
Rogaland		Lysefjorden	Sørside	58.55515	6.11959	2	28/8/2017
Rogaland		Lysefjorden	Sørside	58.58608	6.12354	6	28/8/2017
Rogaland		Lysefjorden	Sørside	58.59338	6.14353	1	28/8/2017
Rogaland		Lysefjorden	Sørside	58.59487	6.14756	1	28/8/2017
Rogaland		Lysefjorden	Sørside	58.59689	6.15576	5	28/8/2017
Rogaland		Lysefjorden	Sørside	58.9844444	6.31444444	1	28/8/2017
Rogaland		Lysefjorden	Sørside	59.0005556	6.31861111	6	28/8/2017

Rogaland	Lysefjorden	Sørside	59.0030556	6.33333333	1	28/8/2017
Rogaland	Lysefjorden	Sørside	59.0036111	6.34472222	7	28/8/2017
Rogaland	Lysefjorden	Sørside	59.0122222	6.41083333	2	28/8/2017
Rogaland	Lysefjorden	Sørside	59.0272222	6.50055556	6	28/8/2017
Rogaland	Lysefjorden	Sørside	59.0288889	6.52833333	1	28/8/2017
Rogaland	Lysefjorden	Sørside	59.0291667	6.53472222	12	28/8/2017
Rogaland	Lysefjorden	Sørside	59.0322222	6.54444444	1	28/8/2017
Rogaland	Lysefjorden	Sørside	59.0352778	6.55583333	2	28/8/2017
Telemark	Stråholmen	Knallaren. Ålekråkeskjær	58.9091667	9.66111111	22	20/8/2016
Telemark	Stråholmen	Rauholmen Ø	58.9052778	9.67166667	27	20/8/2016
Telemark	Jomfruland	Ødegårdskilen	58.8666667	9.58055556	35	20/8/2016
Telemark	Jomfruland	Gjesskjæra	58.875	9.54444444	91	20/8/2016
V-Agder	Farsund	Ystesteinen	58.0213889	6.87888889	2	15/8/2016
V-Agder	Grønnsfjord	Midtfjordøyene	58.06332	7.10081	1	16/8/2016
V-Agder	Lyngdalsfjord	Øst	58.13172	6.91637	7	16/8/2016
V-Agder	Lyngdalsfjord	Vest	58.0566667	6.76805556	1	16/8/2016
V-Agder	Lyngdalsfjord	Vest	58.1236111	6.87555556	12	16/8/2016
V-Agder	Lyngdalsfjord	Vest	58.1469444	6.86361111	1	16/8/2016
V-Agder	Lyngdalsfjord	Vest	58.1719444	6.82611111	7	16/8/2016
V-Agder	Lyngdalsfjord	Vest	58.1525	6.89222222	1	16/8/2016
V-Agder	Mandal	Vestre Ballastskj.	57.9913889	7.79444444	1	17/8/2016
V-Agder	Mandal	Austre Ulveskj.	57.9894444	7.80833333	1	17/8/2016
Vestfold	Horten	Bastøy S	59.3711111	10.5269444	5	19/8/2016
Vestfold	Tjøme	Tjømebåen	59.0230556	10.4127778	89	23/8/2016
Vestfold	Tjøme	Ferder	59.0430556	10.53	31	23/8/2016
Vestfold	Tjøme	V Bustein.Selskj.	59.1263889	120.477778	41	23/8/2016
Vestfold	Nøtterøy	Skarvesetet. Flatskj.	59.1986111	10.5833333	77	23/8/2016
Vestfold	Larvik	Svenner. Suslingane	58.9833333	10.1788889	38	23/8/2016
Vestfold	Larvik	Tvisteinen	58.9433333	9.93222222	2	23/8/2016
Vestfold	Larvik	Stokkøya. Selskjær	59.0030556	9.81944444	14	23/8/2016

Supplemental Appendix II
Harbor Seal Simulation Script

André Moan and Jonas Oliver Elnes

26 January 2021

The following 'Rmarkdown' file displays the packages and script used to simulate the harbor seal (*Phoca vitulina*) distribution from their primary haul-out site to an at-sea location. The simulated sites are then used to estimate the abundance of harbor seals in Statistical Sea Locations (SSLs). The script is divided into four sections: (1) Read in data, (2) Functions, (3) Simulation, and (4) Results. The following 'Rmarkdown' file only displays the script used in the simulation. Explanatory results from the script can be viewed in the Appendix:

Packages used for data management, simulation, and plot:

```
library(sf)  # reading and manipulating spatial data
library(ggplot2) # plotting
library(ggspatial) #plotting
library(rnaturalearth) # administrative borders
library(grid)
library(gridExtra)
library(truncnorm) # truncated normal distribution
library(geosphere) # destPoint function
library(data.table) #data management
```

Data.table is also used for its significantly faster rbindlist function (as an alternative to do.call[rbind, var]) because it can concatenate rows in a dataframe without copying the whole dataframe for each concatenation operation, which would take a lot longer. Also, fread is from data.table and has the same advantage of being much faster.

Read in Data

Get polygons for all the countries in our study region. We need other countries besides Norway because some locations intersect with the coastline for those countries.

```
scandinavia <- ne_countries(country = c("Norway", "Sweden","Finland","Russia"), scale = "large", returnclass="sf")
scandinavia <- st_transform(scandinavia, crs = 4326) # convert to latlong
scandinavia <- st_crop(scandinavia, xmin = 0, xmax = 35, ymin = 55, ymax=72) # crop to region of interest
scandinavia <- st_transform(scandinavia, crs = 2163) # transform back to planar
scandinavia <- st_union(scandinavia) # combine all country polygons into one single big "landmass" polygon
scandinavia <- st_sf(scandinavia)
st_agr(scandinavia) <- "constant"
```

SSLs were obtained from the Institute of Marine Research. SSLs retained for further analysis were preselected in *QGIS*; landmasses were also removed from the polygons. See Figure 2 for an overview of which SSLs were selected. Retained SSLs are here downloaded into *R* and given central coordinates.

```
#coastal SSL and adjacent offshore SSL have been selected in qgis and saved as "SSL coastal 2.0.shp"
locations <- st_read("~/OneDrive - Universitetet i Oslo/master/Q/shp data/SSL coastal 2.0.shp")
# get centers for all cells, but in latlong coordinates
locations[,c("center.X", "center.Y")] <- st_coordinates(st_transform(st_centroid(locations), crs = 4326))
st_agr(locations) <- rep("constant", times = ncol(locations))
```

Read in harbor seal count data retrieved from the IMR and prepare it for the simulation.

```
hss <- fread("~/OneDrive - Universitetet i Oslo/master/R/data/excel csv/telledata.csv", dec=",")
hss$LAT <- as.numeric(hss$LAT)
hss$LON <- as.numeric(hss$LON)
hss <- with(hss, {
  pts <- lapply(seq_len(length(COUNT)), function(i) st_point(c(LON[i], LAT[i])))
  pts <- do.call(st_sfc, args = list(... = pts, crs = 4326))
  st_sf(data.frame(colony = 1:length(COUNT), location = NA, count = COUNT),
```

```
        geometry=st_geometry(pts),
        agr = rep("constant", 3))
})
hss <- hss[!is.na(hss$count),]
hss <- st_transform(hss, crs = 2163) #planar
hss$location <- locate(hss)
hss$colony <- 1:nrow(hss)
st_agr(hss) <- rep("constant", 3)
```

Functions

Functions used in the simulation and data management.

The function mkss() takes an sf object with point coordinates as features and creates a sampling space of discretized points of the given density (e.g., if the sf unit is in meters, a density of 0.001 would correspond to one point per kilometer, and a density of 1 would be one point per meter). Points are constructed by first creating 180 line segments that all originate in the initial point coordinates and end 192.3 km away, each with an angle of 2° offset from the previous one. Points are then created at regular intervals along those lines and named based on the concentric ring they belong to, again originating from the initial point coordinate.

```
mkss <- function(x, density = 0.001) {
  ss <- st_buffer(x, 191300, nQuadSegs = 45) # sampling space
  ss <- st_cast(ss, "POINT")
  ss <- st_union(ss, x, by_feature = T)
  ss <- st_cast(ss, "LINESTRING")
  s2 <- sapply(seq_len(nrow(ss)), function(i) st_line_sample(ss[i,], density=density))
  s2 <- do.call(st_sfc, args=list(...= s2, crs = 2163))
  s2 <- st_sf(data.frame(id = 1:nrow(ss), geometry=s2))
  s2 <- st_cast(s2, "POINT", warn=F)
  s2$ring <- rep(sum(s2$id==1):1, times = max(s2$id))
  s2 <- s2[which(!st_intersects(s2, scandinavia, sparse=F)),]
  s2
}
```

locate() – Determines which location cell a given point falls into.

```
locate <- function(dest) {
   dist <- st_distance(locations, dest)
   j <- apply(dist, 2, which.min)
   locations$LOKREF[j]
}
```

groupify() – Takes a given total number of animals and distributes them into groups so that the numbers of members in the groups are normally distributed with the given mean and SD.

```
groupify <- function(total, mean = 2, sd = 2) {
   lapply(total, function(x) {
      if (x == 1) {
         g = 1
      } else {
         g <- round(rtruncnorm(n = x, a = 1, b = x, mean = mean, sd = sd))
         g <- g[cumsum(g) <= x]
         while (sum(g) != x) {
            b <- x - sum(g)
            if (b > 1) b <- round(rtruncnorm(n = 1, a = 1, b = b, mean = mean, sd = sd))
            if ((sum(g) + b) <= x) g <- c(g, b)
         }
      }
      g
   })
}
```

disperse() – This function takes count data for a colony (x) and a corresponding sf sampling space (ss), and simulates the break-up of a colony into individual groups and dispersal away from the colony site based on the simulation parameters. The function is vectorized, so x and ss can both be data frames. Returns an sf object with the new positions for each group, the number of individuals in that group, and the fishery statistics cell to which that new position corresponds.

```
disperse <- function(x, ss, probs) {
  g <- groupify(x$count, mean = group_size_mean, sd = group_size_sd)[[1]]
  direction <- sample(unique(ss$id), size = length(g), replace=T)
  s <- lapply(direction, function(dir) {
    i <- which(ss$id == dir)
    if (length(i)==1) return(ss[i,])
    p <- probs[ss$ring[i]]
    if (sum(p) == 0) {
      p <- rep(1/length(i), length(i))
    } else {
      p <- p/sum(p)
    }
    j <- sample(i, size = 1, replace=T, prob = p)
    ss[j,]
  })
  s <- st_sf(do.call(rbind, s))
  s$colony <- x$colony
  s$count <- g
  s$location <- locate(s)
  s
}
```

sim_get_probs() – Convenience function to drop impossible dispersal probabilities. Needed to handle islands, fjords, and jagged coastlines in the sampling space.

```
sim_get_probs <- function(ddm, dds) {
 sapply(c(0, seq(500, 191300, 1000)), function(i) {
  j <- ifelse(i == 0, 500, 1000)
  ptruncnorm(q = i+j, a = 0, b = dispersal_distance_max, mean = ddm, sd = dds)-
   ptruncnorm(q = i, a = 0, b = dispersal_distance_max, mean = ddm, sd = dds)
 })
}
```

sim_run() – Function for running the simulation:

```
sim_run <- function(season, N = 1) {
  ddm <- dispersal_distance_mean[season]
  dds <- dispersal_distance_sd[season]
  p <- sim_get_probs(ddm, dds)
  x <- replicate(N, mapply(disperse, hss.list, ss, MoreArgs = list(probs = p), SIMPLIFY=F), simplify=F)
  setattr(x, “N”, N)
  setattr(x, “ddm”, ddm)
  setattr(x, “dds”, dds)
  x
}
```

sim_run_batch – Runs the simulation in batches of some number of replicates as specified in saveFreq (e.g., 50) and saves all progress by the end of each batch.

```
sim_run_batch <- function(N = 1000, saveFreq = 2, savePath) {
  if (!file.exists(savePath)) stop(sprintf(“Path ‘%s’ not found.”, path))
  if (saveFreq > 100) warning(“A save frequency greater than 100 may cause data loss and is not recommended.”)
  savePath <- gsub(“/$”, “”, “~/OneDrive - Universitetet i Oslo/master/R/data/sim”)
```

```
  progress <- file.path(savePath, "progress.Rdata")
  data <- file.path(savePath, "step%d.RData")

  if (file.exists(progress)) {
    step <- readRDS(progress) + 50
    print(sprintf("Restarting simulation from step %d (If you would like to restart the simulation, please delete the progress file   '%s')",
           step, progress),quote=F)
  } else {
    step <- 1
    print("Starting a new simulation...")
  }

  while (step <= N) {
    print(sprintf("Running replicates %d - %d... ", step, step+saveFreq-1))
    pb <- txtProgressBar(min = 0, max = 12, style = 2)
    res <- lapply(c(season = 1:3), function(season) {
      x <- sim_run(season = season, N = saveFreq)
      setTxtProgressBar(pb, season)
      return(x)
    })
    close(pb)
    saveRDS(res, file = sub("%d", step, data, fixed=T))
    saveRDS(step, file = progress)
    cat(sprintf(" done! Progress saved in '%s'\n", sub("%d", step, data, fixed=T)))
    step <- step + saveFreq
  }
}
```

Simulation

The simulation uses parameters defined in the script to simulate dispersion of harbor seals at sea from their location of molt. The parameter used is the mean travel distance to an at-sea location with standard deviation (SD).

Set the mean and SD for harbor seal average distances from haul-out site to an at-sea location. The group size was set to 0.01 to represent solitary foraging behavior.

```
# set up the simulation parameters

#mean
dispersal_distance_mean <- c(5000, # summer
                       15000, # spring & autumn
                       35000) # winter

#sd
dispersal_distance_sd <- c(11600, # summer
                     26900, # spring & autumn
                     49300) # winter

dispersal_density <- 0.001 # sampling space resolution (1 = 1m, 0.001 = 1km)
dispersal_distance_max = Inf # maximum dispersal distance (meters)
group_size_mean = 1 # average group size
group_size_sd = 0.01 # standard deviation of group size
N <- 1000 # number of simulation replicates (as many as possible, and at least 1000)
```

Set up the sampling space for each colony.

```
hss.list <- split(hss, 1:nrow(hss))
ss <- lapply(hss.list, mkss, density = dispersal_density)
```

Run the simulation in batches.

```
sim_run_batch(saveFreq = 50, savePath = "~/OneDrive - Universitetet i Oslo/master/R/data")
```

The Results are now ready but need to be combined into one large dataset. Combine the batches saved by savefrequenzy. The savefrequenzy was set to 50, and a total of 20 batches of R-objects need to be combined into one large R-object. Each R-object with 50 iterations are 2.4 GB and are added together five at a time. Then the workspace is deleted to add the next five batches together. The workspace should be deleted so that the computer's RAM are not overloaded.

```
# combine the results into one list
dat1 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/step1.Rdata") #9.226s
dat2 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/step21.Rdata")
dat3 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/step41.Rdata")
dat4 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/step61.Rdata")
dat5 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/step81.Rdata")
dat6 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/step101.Rdata")
dat7 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/step121.Rdata")
dat8 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/step141.Rdata")
dat9 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/step161.Rdata")
dat10 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/step181.Rdata")
res1av5 <- do.call(mapply, args = list(c, dat1, dat2, dat3, dat4, dat5, dat6, dat7, dat8, dat9, dat10,  SIMPLIFY = F)) saveRDS(res1av5, "~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/res1av5.Rdata")
rm(list=ls()) #clear workspace

# This procedure was repeated to include all samples.

res1av5 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/res1av5.Rdata")
res2av5 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/res2av5.Rdata")
res3av5 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/res3av5.Rdata")
res4av5 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/res4av5.Rdata")
res5av5 <- readRDS("~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/res5av5.Rdata")

#Combine into one large file
res <- do.call(mapply, args = list(c, res1av5, res2av5, res3av5, res4av5, res5av5, SIMPLIFY = F)) #res4av5, res5av5,
saveRDS(res, "~/OneDrive - Universitetet i Oslo/master/R/data/sim/sim/res.Rdata")
```

Results and Plots

sim_calc_averages – Calculates the mean and SD of simulated harbor seals for each SSL. The function also puts harbor seals simulated outside the boundaries of the retained SSLs used to the closest SSLs of the simulated point. Results are plotted by the later function 'plot_abundance_by_season.'

```
sim_calc_averages <- function(x, collapse = TRUE) {
  R <- 1:length(x[[1]])
  y <- cbind(data.frame(location = locations$LOKREF),
        iter=matrix(0, ncol = length(R), nrow = nrow(locations)))
  y2 <- vector(mode = "list", length = length(R))
  y2 <- lapply(1:3, function(season) cbind(data.frame(season = season), y))

  nil <- sapply(1:3, function(season) {
    sapply(seq_len(length(x[[season]])), function(iter) {
      m <- rbindlist(x[[season]][[iter]])
```

```
      m <- m[,.(count = sum(count)), by = location]
      j <- match(m$location, y2[[season]]$location)
      y2[[season]][j,iter+2] <<- m$count
    })
  })
  y2 <- rbindlist(y2)
  i <- seq(3, 2+length(R), 1)
  y2$mean <- rowMeans(y2[,i,with=F], na.rm=T)
  y2$sd <- apply(y2[, i, with=F], 1, sd, na.rm=T)
  y2$lower <- pmax(0, y2$mean + y2$sd * qnorm(0.025))
  y2$upper <- y2$mean + y2$sd * qnorm(0.975)
  if (collapse == TRUE) {
    y2[,.SD,.SDcols = c(1:2,(ncol(y2)-3):(ncol(y2)))]
  } else {
    y2
  }
}

res.avg <- sim_calc_averages(res)
saveRDS(res.avg, "~/R/res.avg.Rdata")
res.avg <- readRDS("~/R/res.avg.Rdata")
```

plot_dispersal() – Plots a selected replica of a given colony in the sampling space defined for the colony.
plot_dispersal2() – plots a selected replica of a given colony in the sampling space by the seasonal distances used. Results can be inspected in Appendix 1, Figure I.1.

```
plot_dispersal <- function(x, replicate = 1, colony, months = 1:3) {

  d <- lapply(months, function(month) {
    cbind(data.frame(replicate = replicate, colony = colony, month = month),
       res[[month]][[replicate]][[colony]])
  })
  d <- rbindlist(d)
  d <- st_as_sf(d)
  ggplot(data = d) +
    geom_sf(data = ss[[colony]], shape = 21, fill = "white", color = "black", alpha = 0.25) +
    geom_sf(data = d, aes(size = count), shape=21, color="white", fill="firebrick", alpha = 0.75) +
    geom_sf(data = hss[colony,], colour = "steelblue2") +
    coord_sf(crs = 4326)
}

plot_dispersal2 <- function(x, replicate, colony, months = 1:3) {
  len_r <- length(x[[1]])
  if (replicate > len_r) stop(sprintf("Max replicate is %d", len_r))

  d <- lapply(months, function(month) {
    cbind(data.frame(replicate = replicate, colony = colony, month = month),
       res[[month]][[replicate]][[colony]])
  })
  d <- rbindlist(d)
  d <- st_as_sf(d)
  d$month <- factor(d$month, labels = c("Su", "S&A", "Wi")) #Wi, S&A, Su #month.name[1:3]
  bb <- st_bbox(st_transform(st_buffer(hss[colony,], 60000), crs = 4326))

  ggplot(data = d) +
    geom_sf(data = scandinavia) +
    geom_sf(data = locations, fill = NA) +
    geom_sf(data = d, aes(size = count), shape=21, color="white", fill="firebrick", alpha = 0.75) +
```

```
    geom_sf(data = hss[colony,], shape = 21, size = 3, fill  = "steelblue2", color="black") +
    coord_sf(xlim=bb[c(1,3)], ylim = bb[c(2,4)], crs = 4326, expand=T) +
    theme_bw() +
    theme(plot.background = element_rect(fill="white")) +
    facet_wrap(~month)
}

#plot
#first replica of colony 4
a <- plot_dispersal(x = res.avg, colony = 4, replicate = 1)
a1 <- plot_dispersal2(x = res, replicate = 1, colony = 4, months = 1:3)
grid.arrange(a, a1, ncol = 1) #Plot both plots in one frame, see figure A.1

#second replica of colony 4
b <- plot_dispersal(x = res.avg, colony = 4, replicate = 2)
b1 <- plot_dispersal2(x = res, replicate = 2, colony = 4, months = 1:3)
grid.arrange(b, b1, ncol = 1) #Plot both plots in one frame, see Figure A.1
```

plot_abundance_by_season – Plot results with the mean and SDs for every SSL with simulated harbor seals in them. Results can be inspected in Appendix 1, Figure I.2.

```
plot_abundance_by_season <- function(x, area) {
  i <- grep(sprintf("^%02s", area), x$location)
  if (!length(i)) stop("Area not in data")

  d <- x[i,][!is.nan(mean),]
  valids <- d[,.(valid=sum(mean)!=0), location]
  valids <- valids$location[valids$valid]
  d <- d[location %in% valids,]
  title <- sprintf("Simulated location use through an average year in area %02s (w/%d replicates)", area, attr(x, "N"))

  ggplot(data = d, aes(x = month, y = mean)) +
    geom_ribbon(aes(ymin = lower, ymax = upper), fill="steelblue2", alpha = 0.25) +
    geom_point(color = "firebrick") +
    geom_line(color = "firebrick") +
    #expand_limits(y = 0) +
    ylim(c(0, NA)) +
    ylab("Number of seals") +
    scale_x_continuous("Season", breaks = c(1, 2, 3), labels = c("Su", "S&A", "Wi")) +
    ggtitle(title) +
    facet_wrap(~location, scales = "free_y")
}

#plot
plot_abundance_by_season(x = res.avg, area = 0)
# This procedure was repeated to include all Statistical Sea Locations.
```