Høgskolen i **Hedmark**

The bivariate Power-Normal and the bivariate Johnson's System bounded distribution in forestry, including height curves. Supplementary material.

Title:

The bivariate Power-Normal and the bivariate Johnson's System bounded distribution in forestry, including height curves. Supplementary material.

Author:

Erik Mønness Hedmark University College, ØLR P.O. Box 104, 2450 Rena, Norway Phone: +47 62430598 Fax: +47 62430500 e-mail: erik.monness@hihm.no

Number: 9	**Year:** 2014	**Pages:** 80	**ISBN:** 978-82-7671-958-1 **ISSN:** 1501-8563

Financed by:

Keywords:

Bi-normal, bivariate Johnson's System bounded distribution, bivariate power-normal distribution, height curve, Box-Cox transformation.

Content

Summary:

This document contain supplementary material concerning two published papers:

Mønness, E. 2011b. *The Power-Normal Distribution: Application to forest stands*. Canadian Journal of Forest Research 41(4): 707-714. doi: 10.1139/X10-246.

Mønness, E. (2014). The bivariate power-normal distribution and the bivariate Johnson system bounded distribution in forestry, including height curves.

Canadian Journal of Forest Research, 45(3), 307-313. doi: 10.1139/cjfr-2014-0333

Supplementary material to the above article; SAS programs and computing details are found in

Mønness, E. 2011a. *The Power-Normal Distribution and Johnsons System bounded distribution: Computing details and programs.* Available from http://hdl.handle.net/11250/133513 3] .

From published abstracts:

The Power-Normal (PN) distribution, originated from the inverse Box-Cox transformation, is presented and some possibilities in forest research are explored. The Power-Normal achieve shapes, by a Skewness * Kurtosis value, common to diameters and heights of forest stands. The estimation of the parameters by maximum likelihood is straightforward with good numerical properties. The shapes achieved by PN are very diverse even with only three parameters: The Johnson System bounded distribution (SB), also used in forestry, can encounter numerical problems with maximum likelihood estimation. The PN distribution is seen to give good estimates of diameter and height distributions, judged by the Kolmogorov-Smirnov statistic and visual inspection. It seems to perform better than the SB, especially on heights.

A bivariate diameter and height distribution yields a unified model of a forest stand. The bivariate Johnson's System bounded distribution and the bivariate power-normal distribution are explored. The power-normal originates from the well-known Box-Cox transformation. As evaluated by the bivariate Kolmogorov-Smirnov distance, the bivariate power-normal distribution seems to be superior to the bivariate Johnson's System bounded distribution.

The conditional median height given the diameter is a possible height curve and is compared with a simple hyperbolic height curve. Evaluated by the height deviance, the hyperbolic function yields the best height prediction. A close second is the curve generated by a bivariate power-normal distribution. Johnson's System bounded distributions suffer from the sigmoid shape of the association between height and diameter.

The data is from

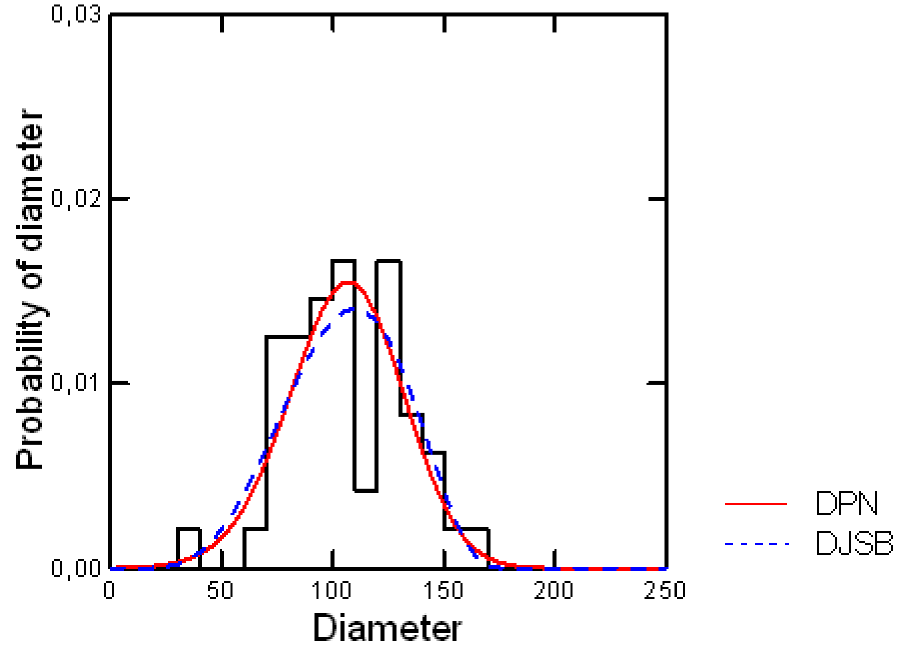
Vestjordet, E. 1977. Precommercial thinning of young stands of Scots Pine and Norway Spruce I: Data stability, dimension distribution etc. Medd. Nor. inst. skogforsk 33(9): 1-436.

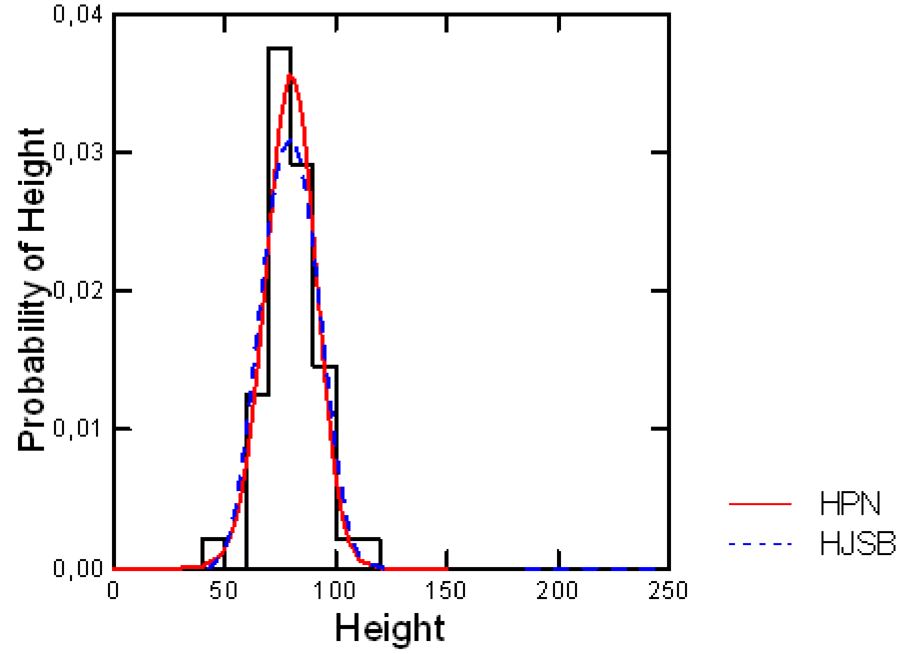
Plots of all stands

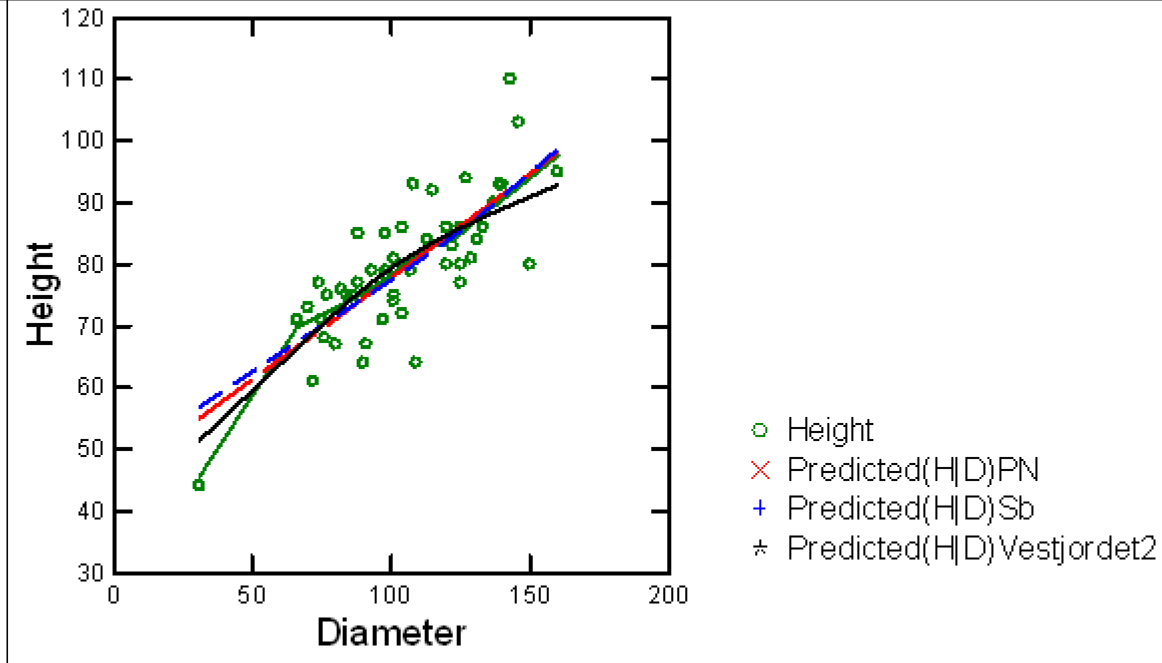
	Histogram Diameter (mm)	Histogram Height (dm)	Plot Diameter (mm) * Height (dm)
PowerNormal	RED	RED	RED
Johnson System Bounded	BLUE	BLUE	BLUE
Observed histogram grouped in units of 10	BLACK	BLACK	
Vestjordet2 height curve			BLACK
Individual trees with LOWESS regression			GREEN
	The curves is a LOWESS based on point estimates of the density and may sometimes appear strange. The scales are fixed, equal in all figures.		The scales vary dependent of actual data values.

The histograms are marginal diameter and height distributions. The plots show diameter/height relation with estimated height given diameter curve.

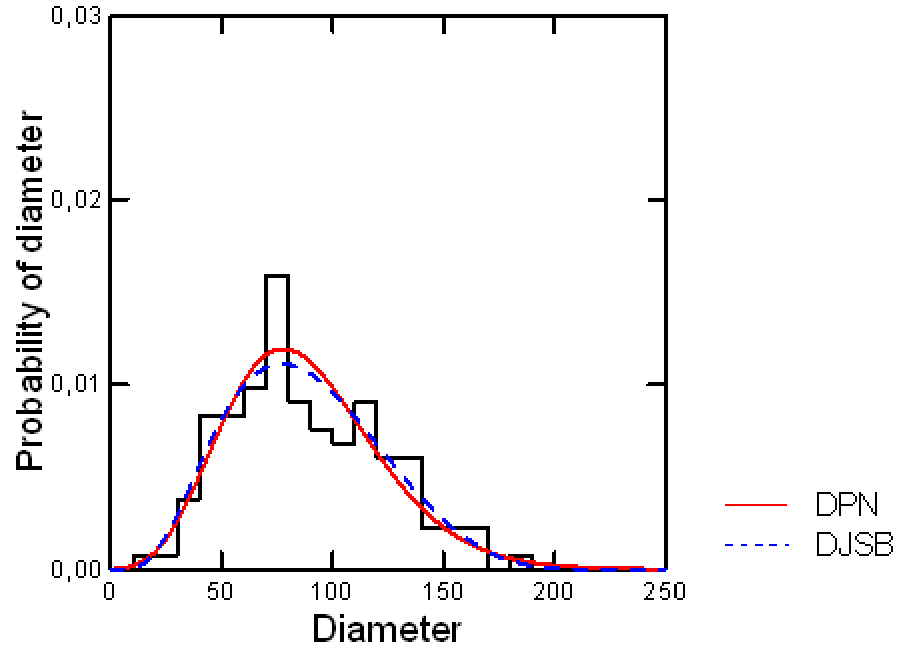
Results (↓) for STAND = 1

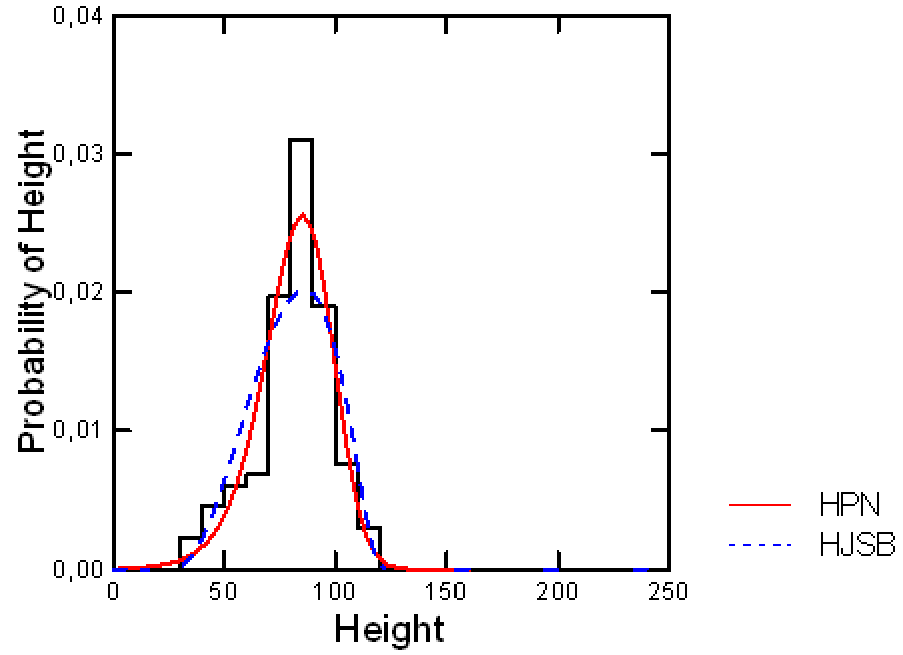


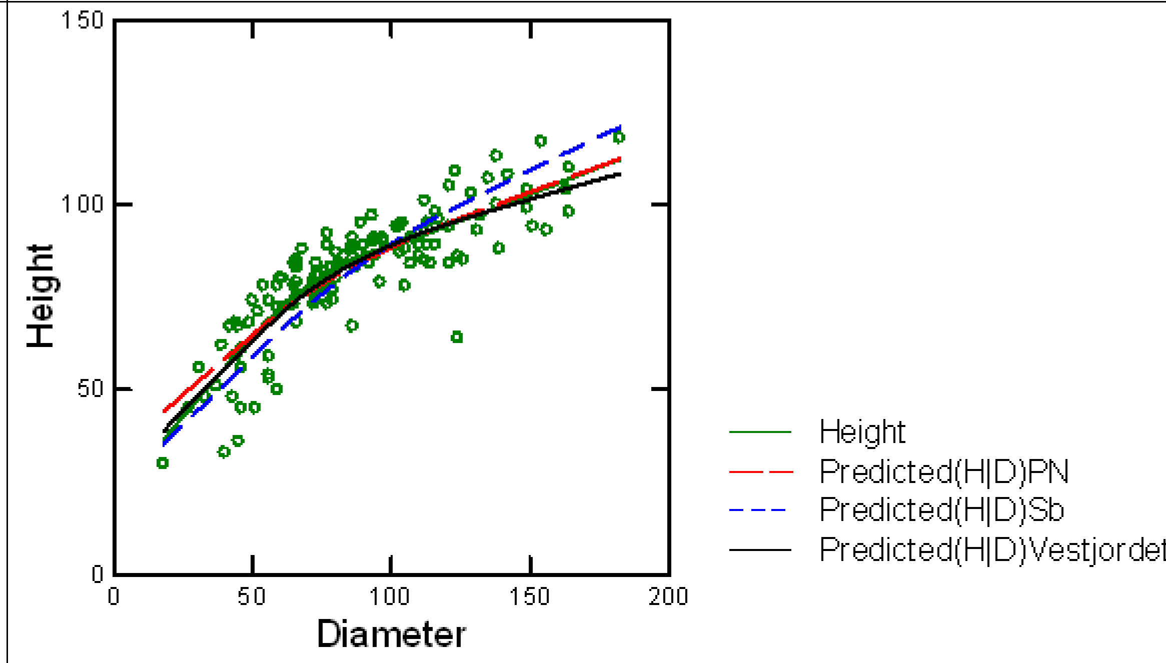




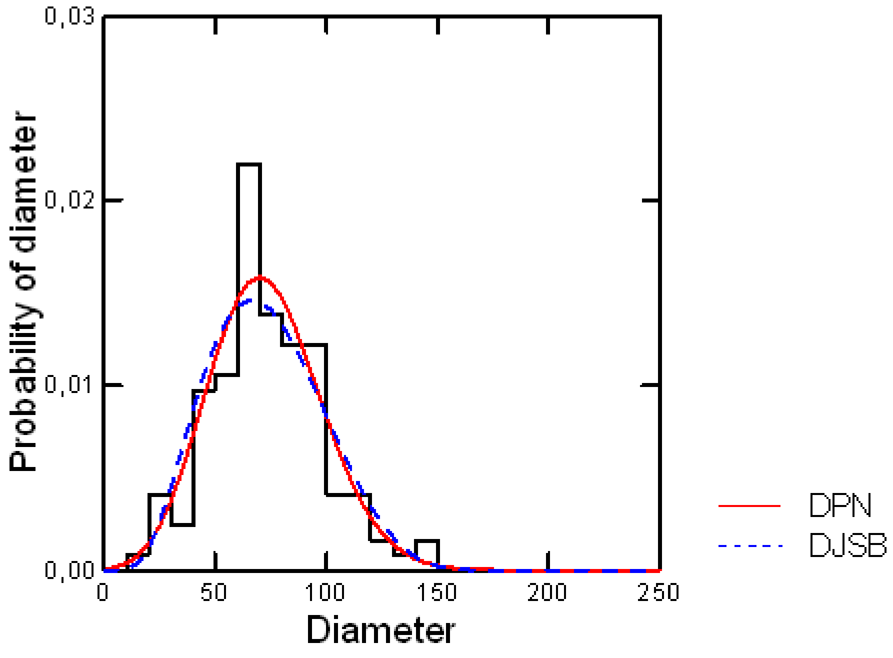
Results (↓) for STAND = 2

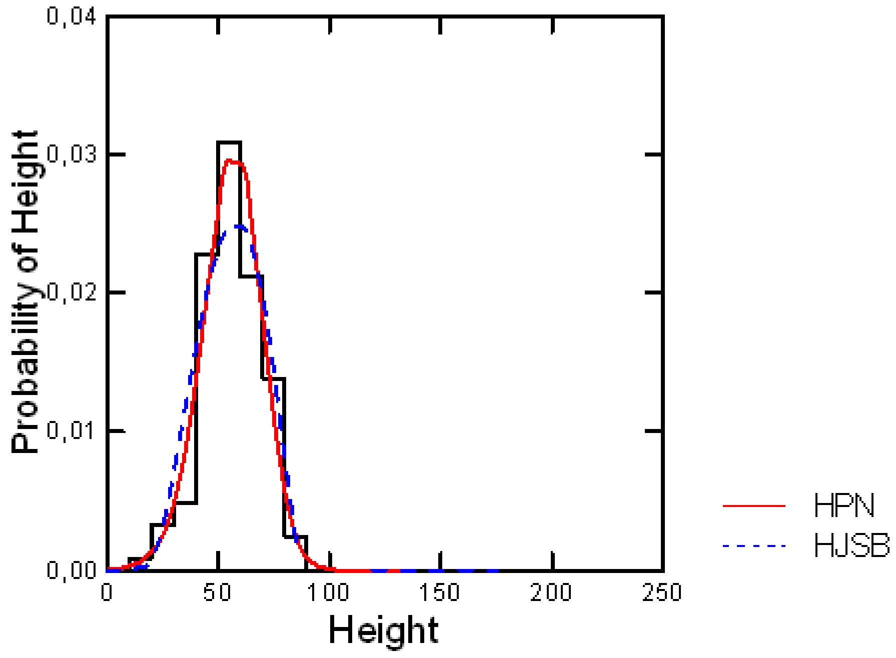


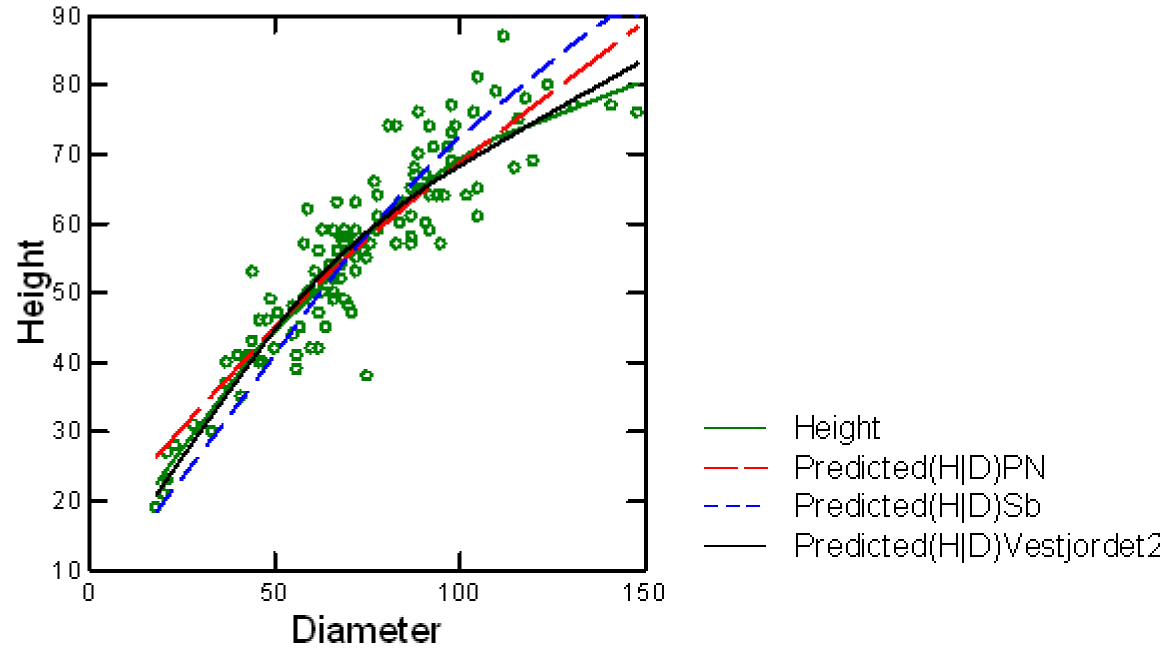




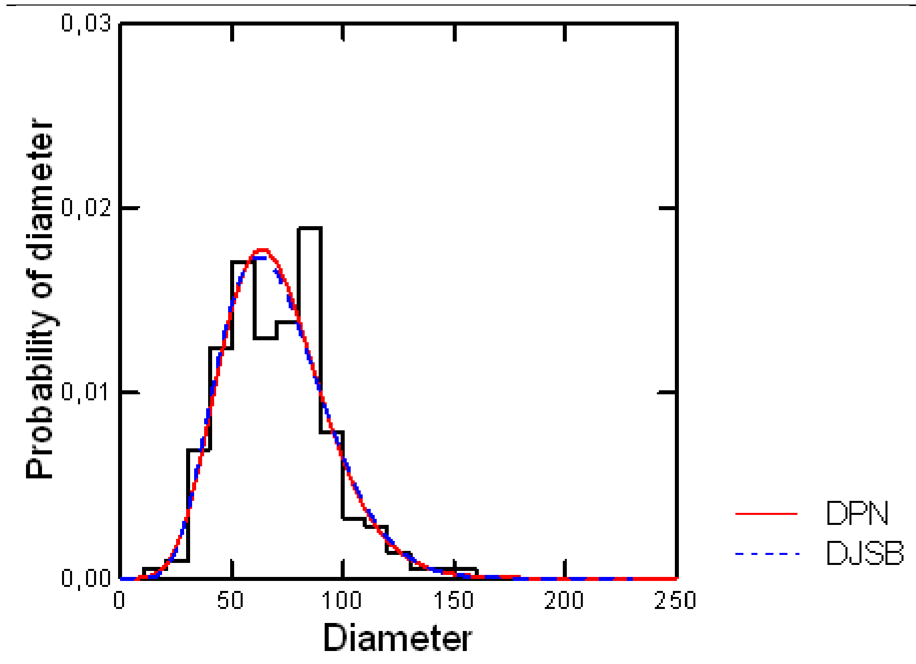
Results (↓) for STAND = 3

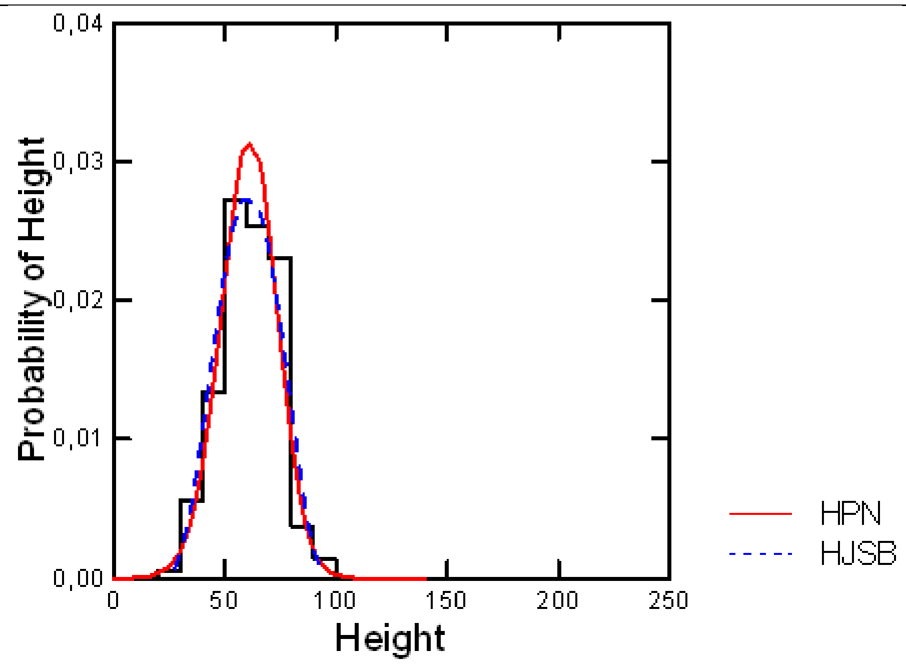


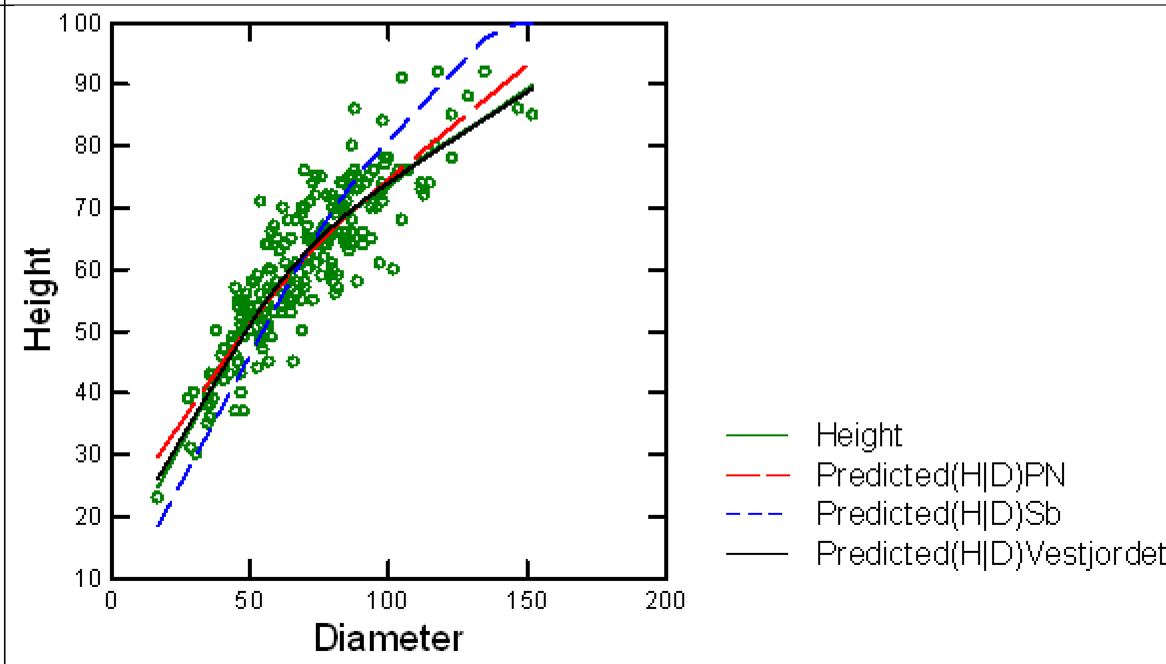




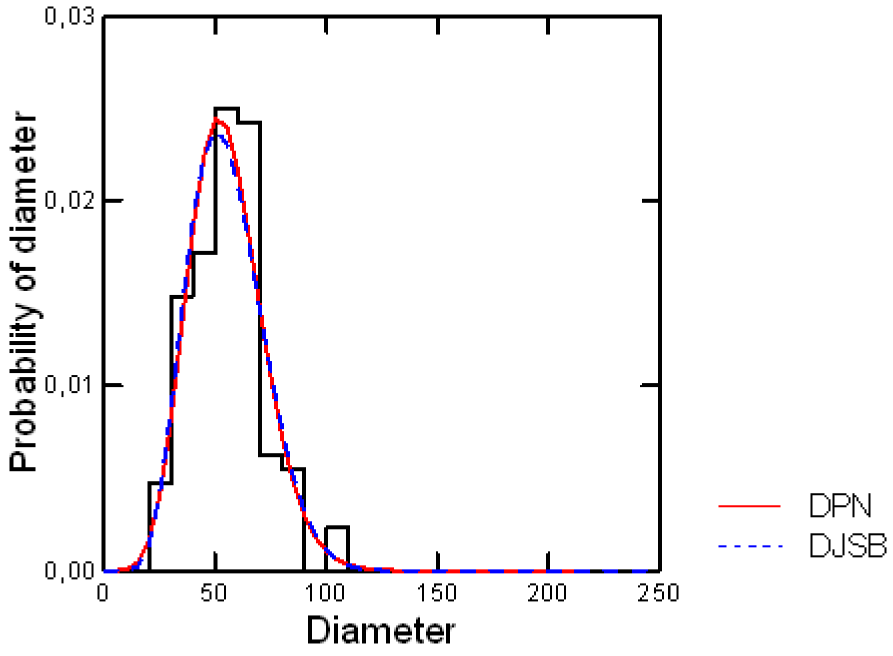
Results (↓) for STAND = 4

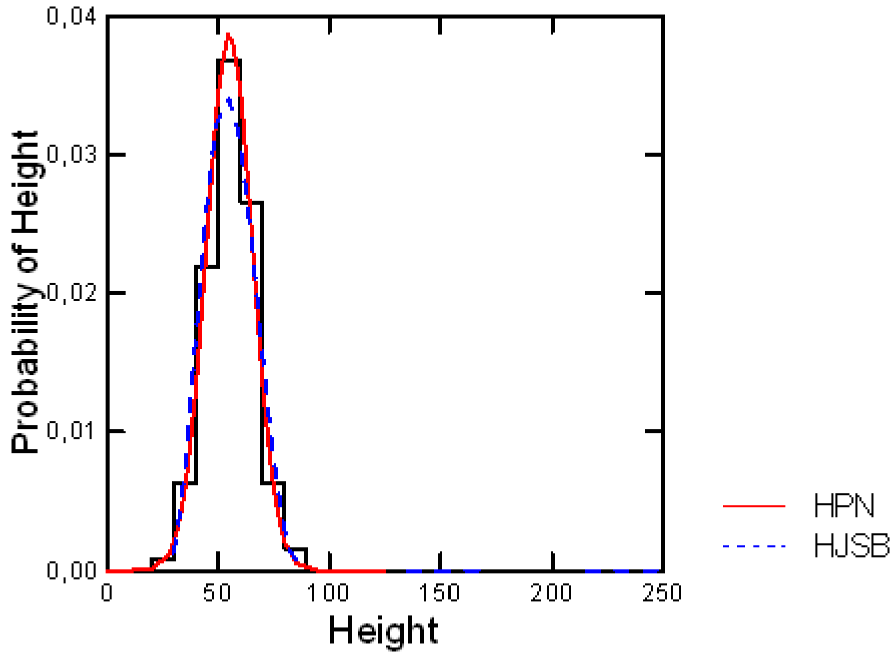


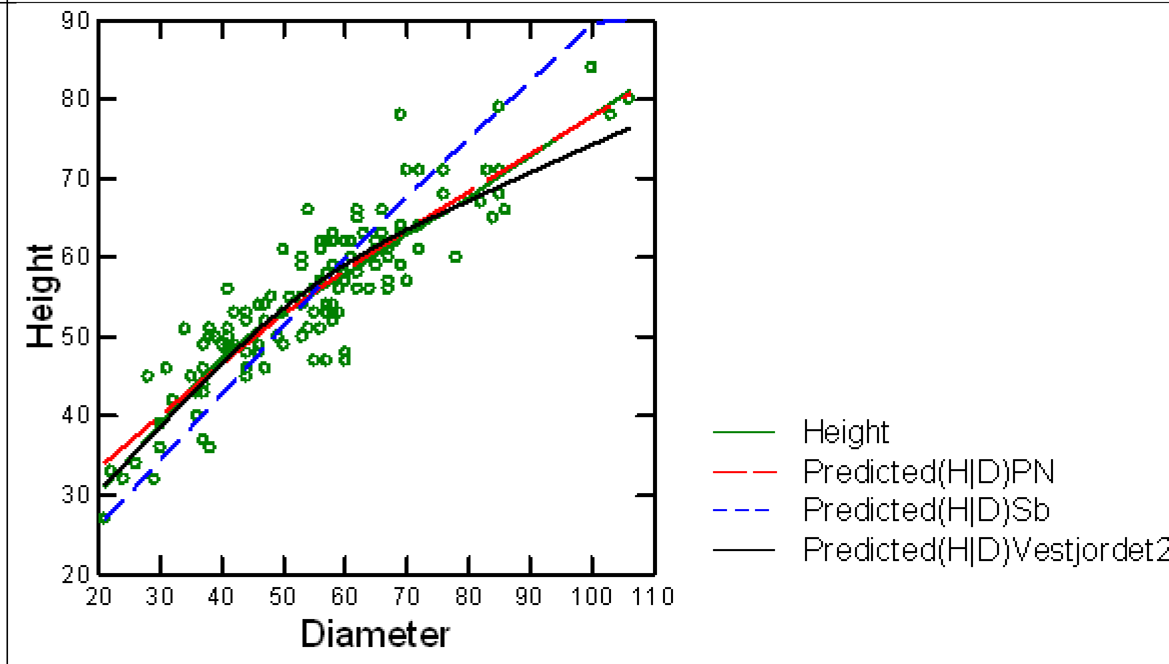




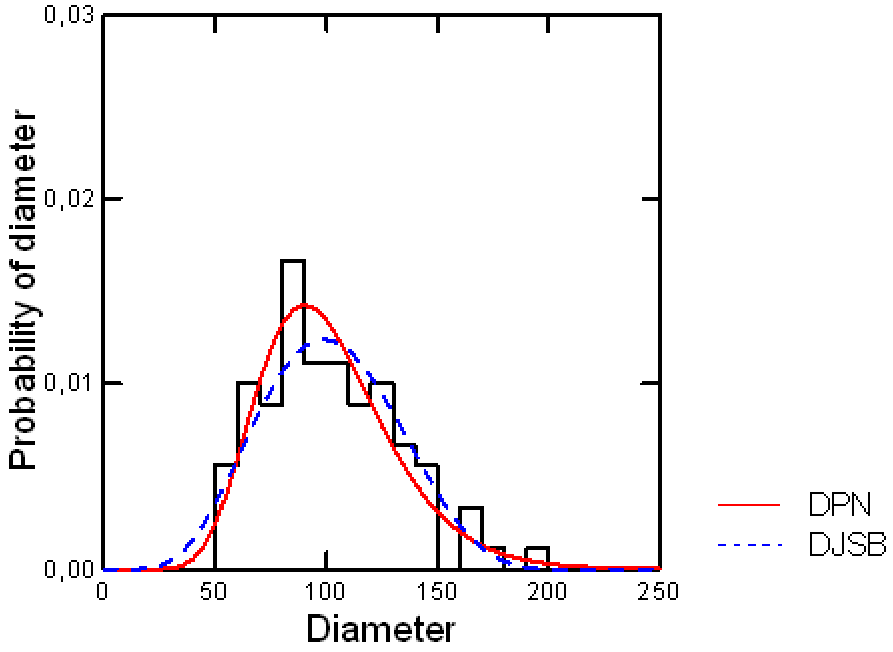
Results (↓) for STAND = 5

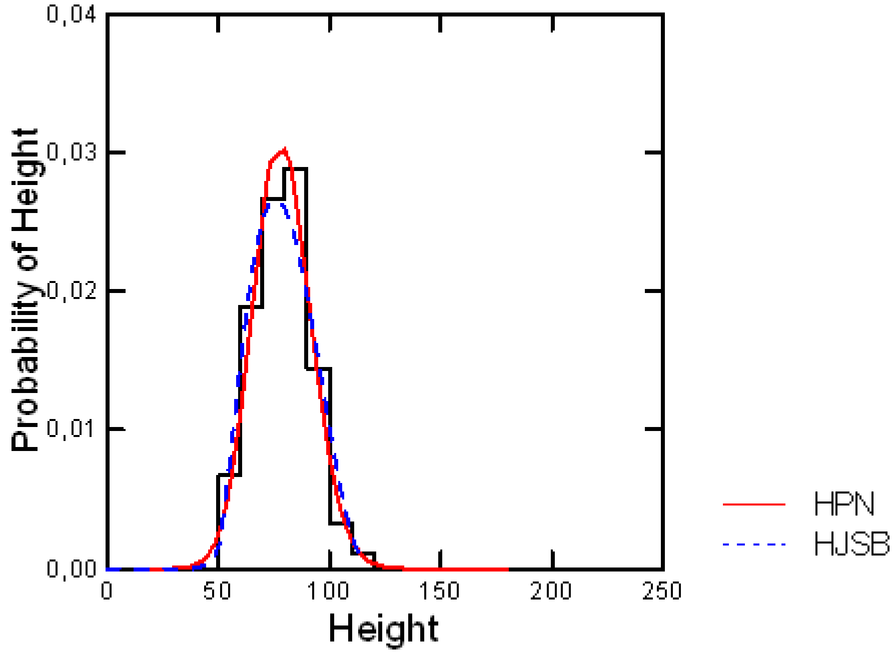


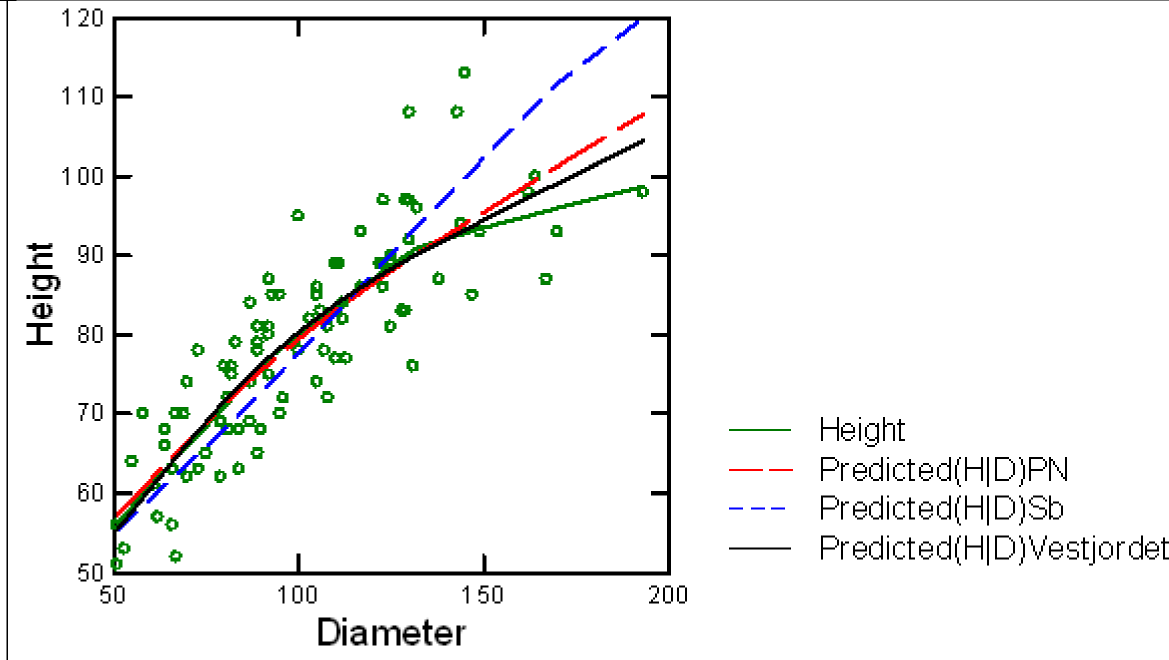




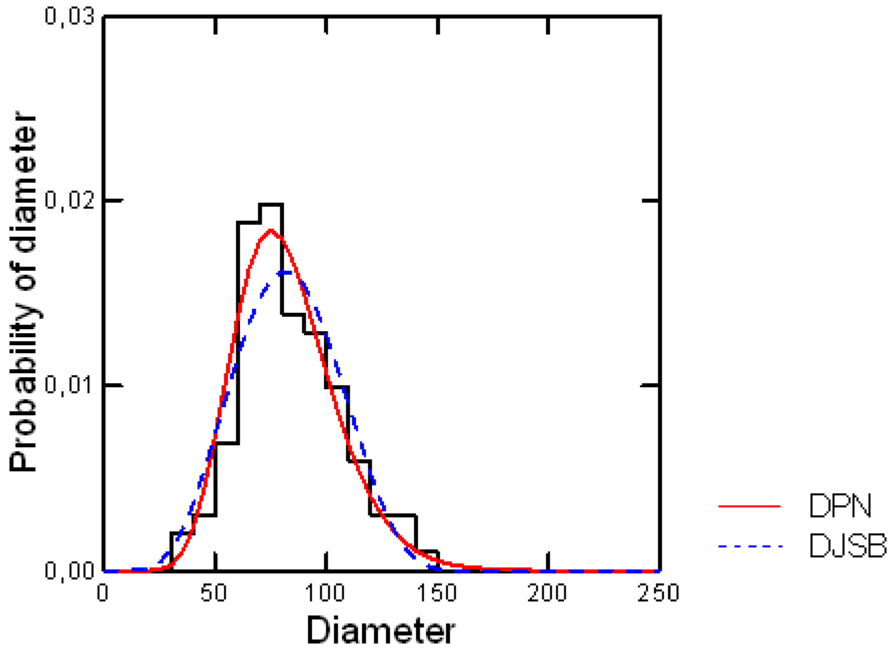
Results (↓) for STAND = 6

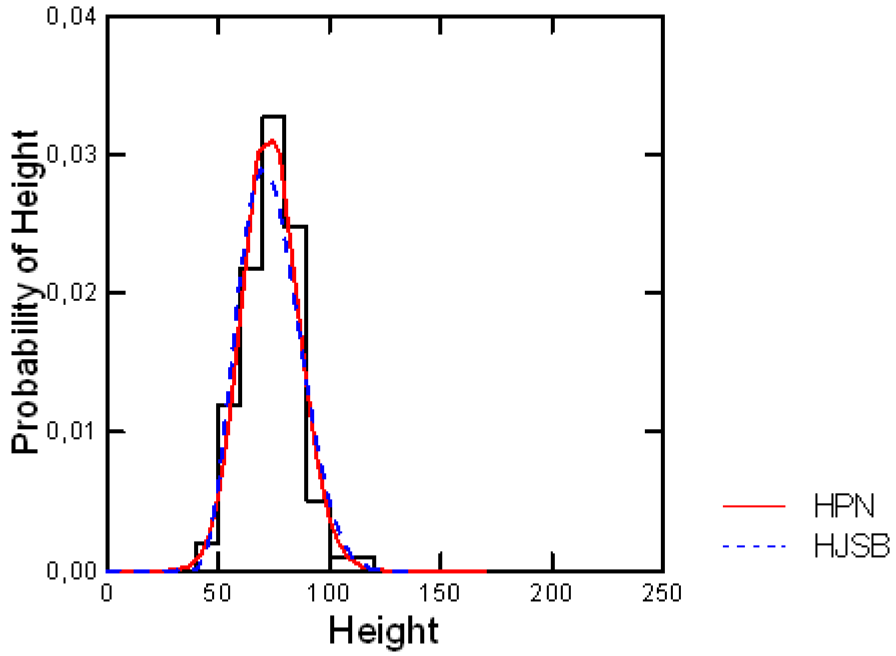


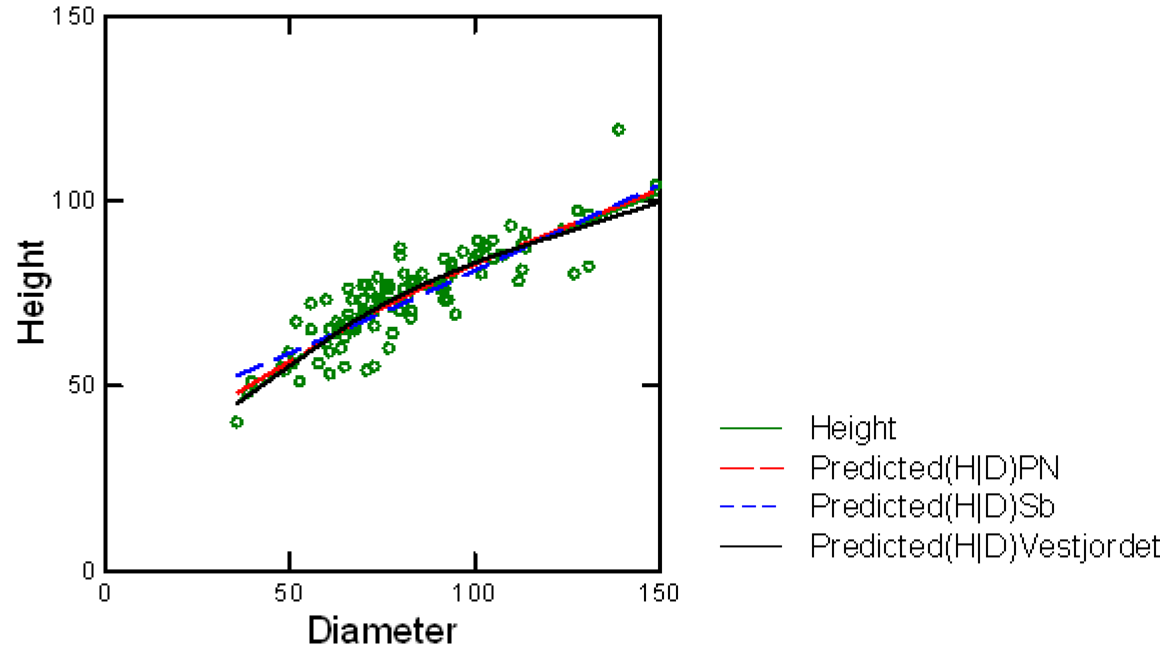




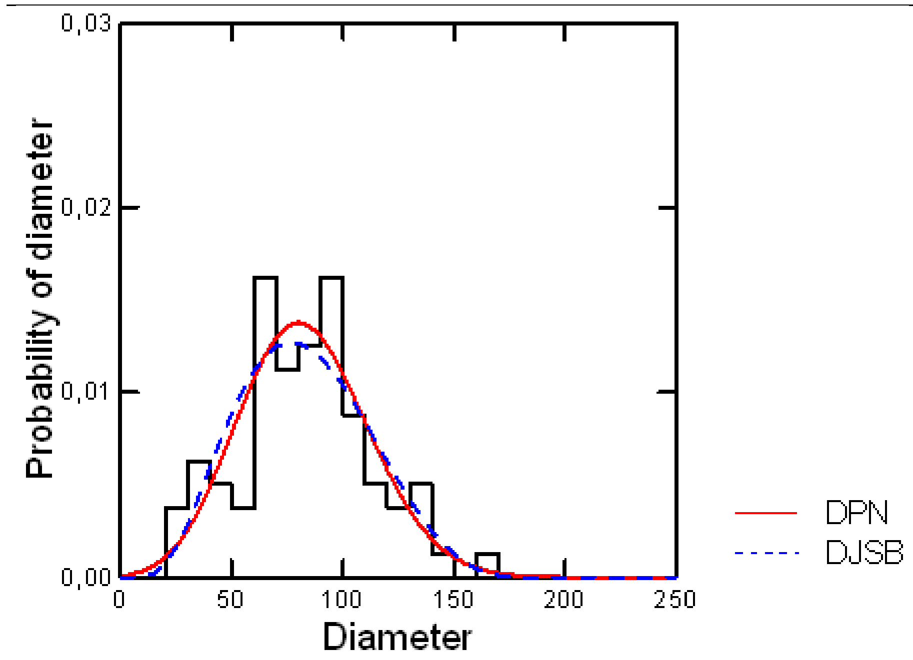
Results (↓) for STAND = 7

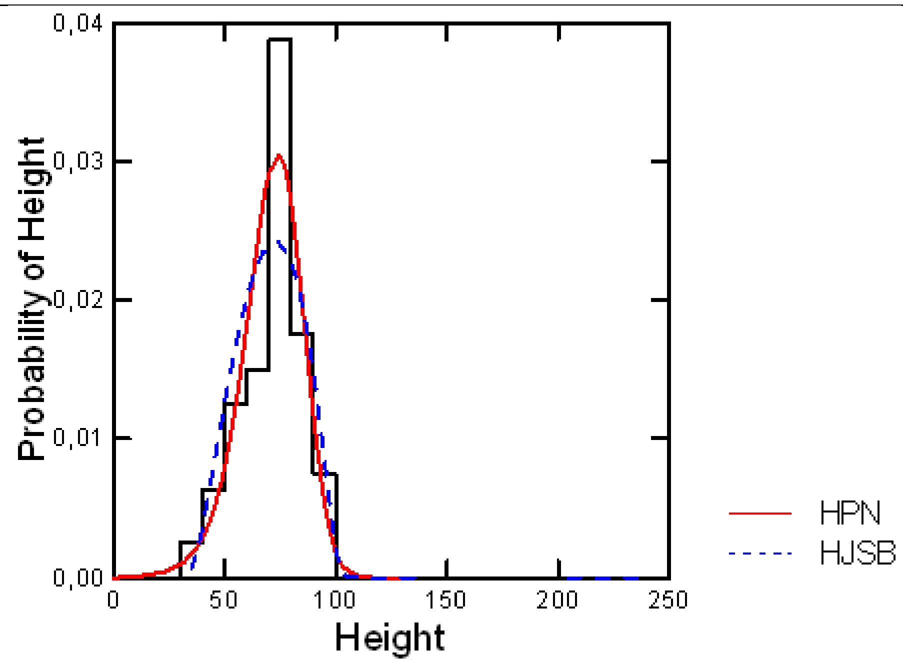


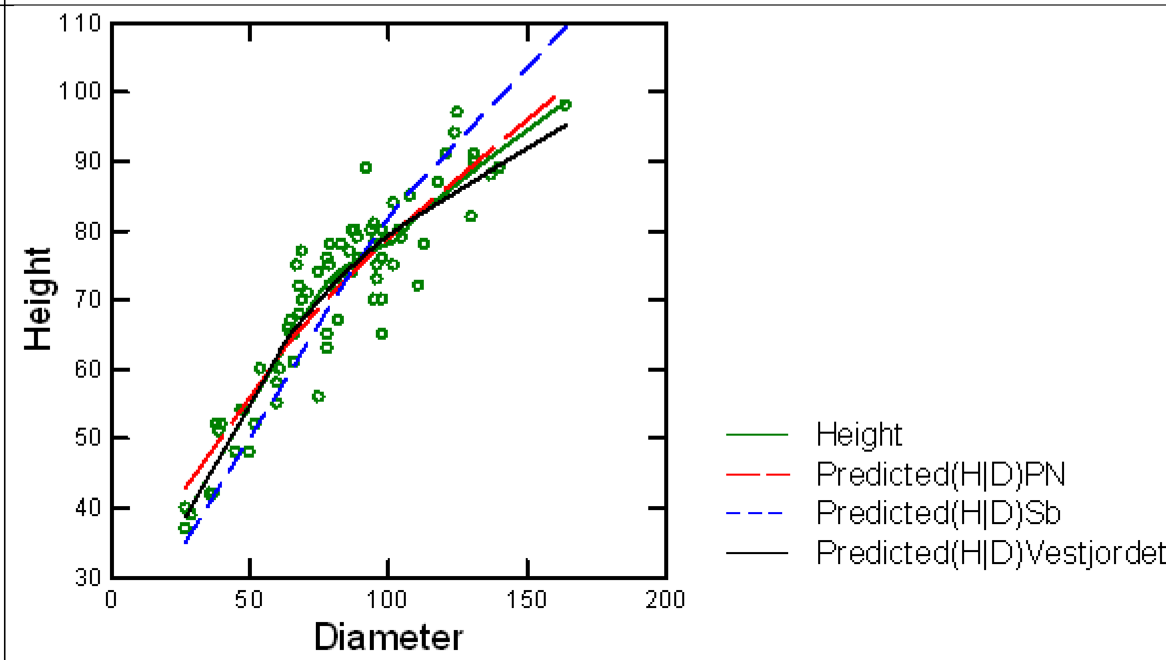




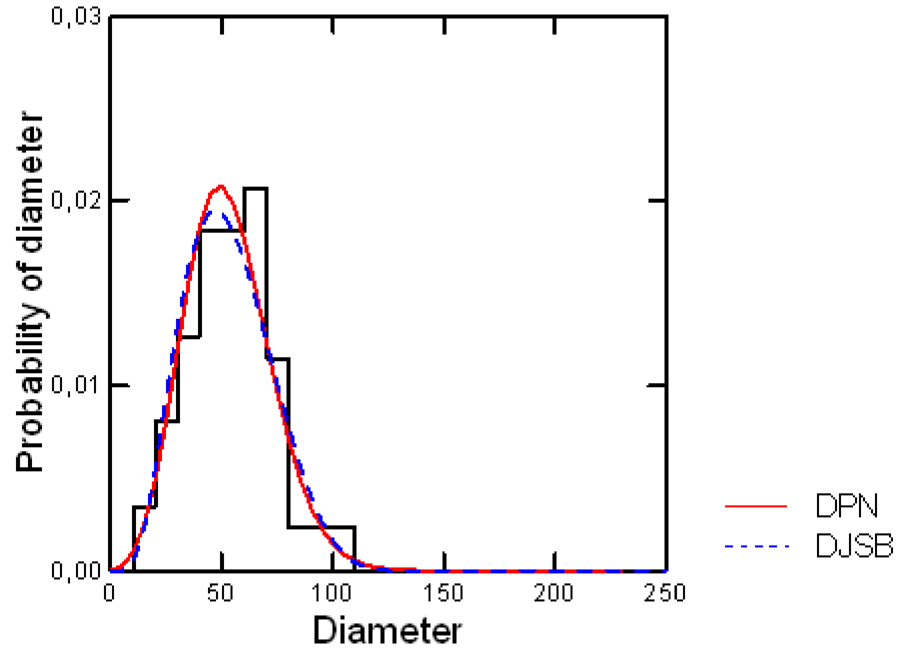
Results (↓) for STAND = 8

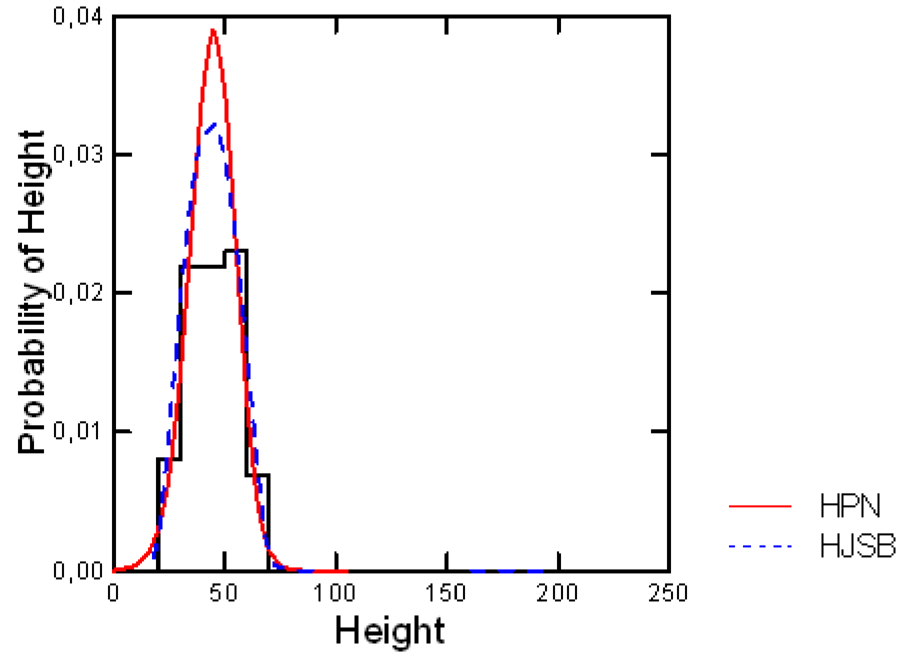


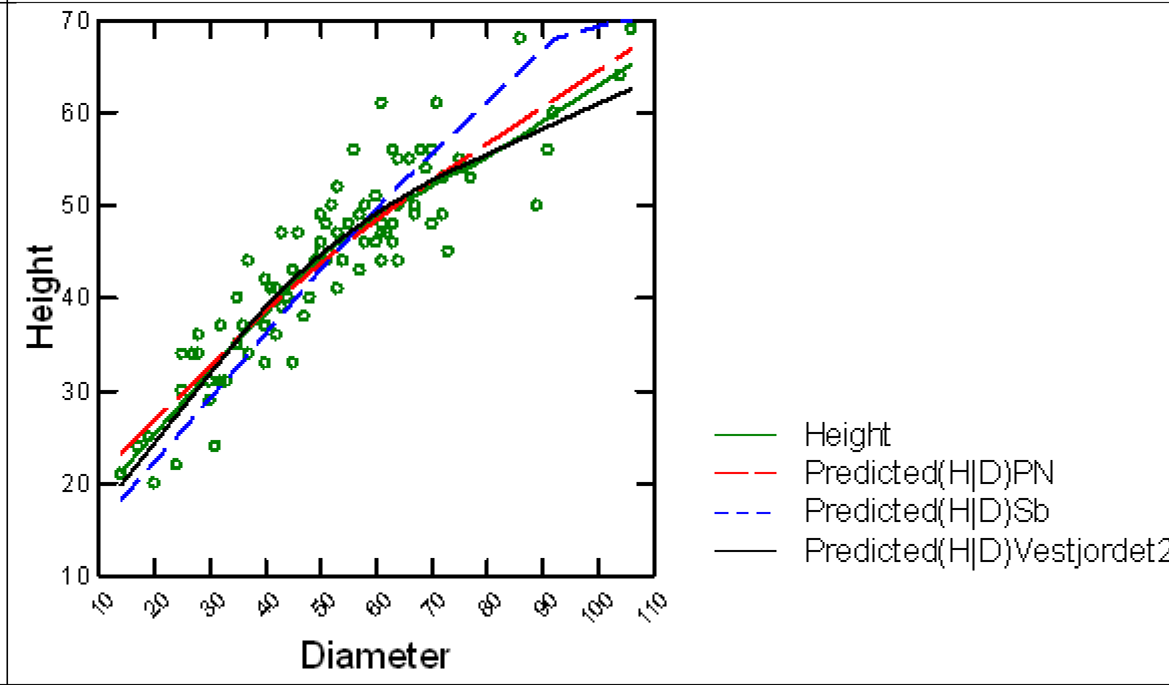




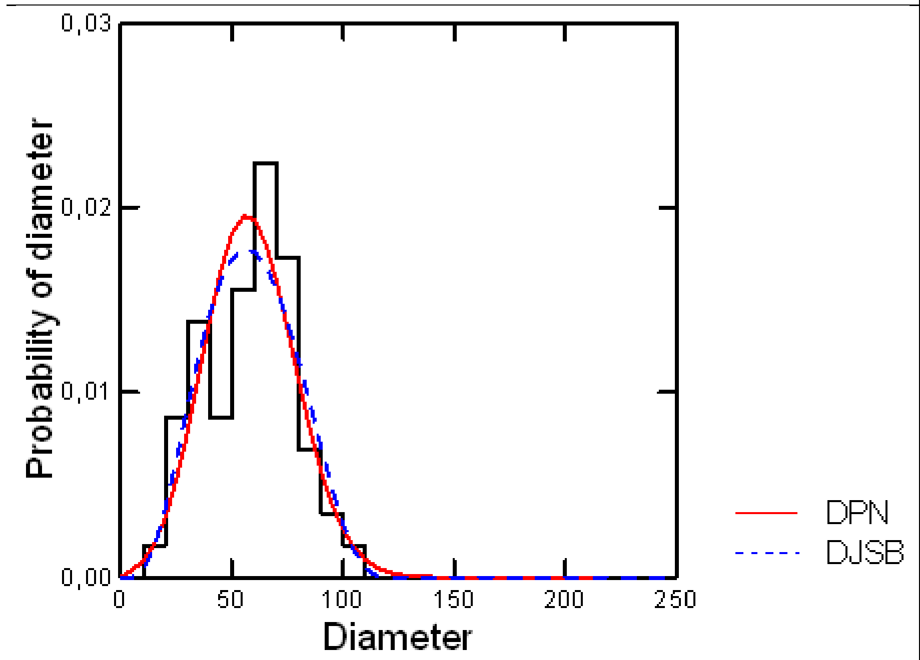
Results (↓) for STAND = 9

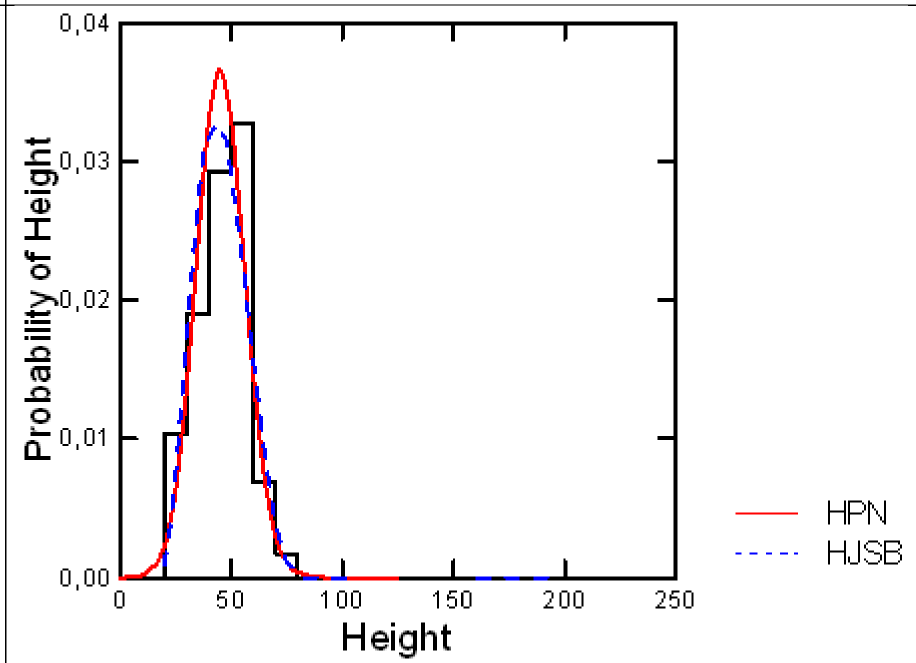


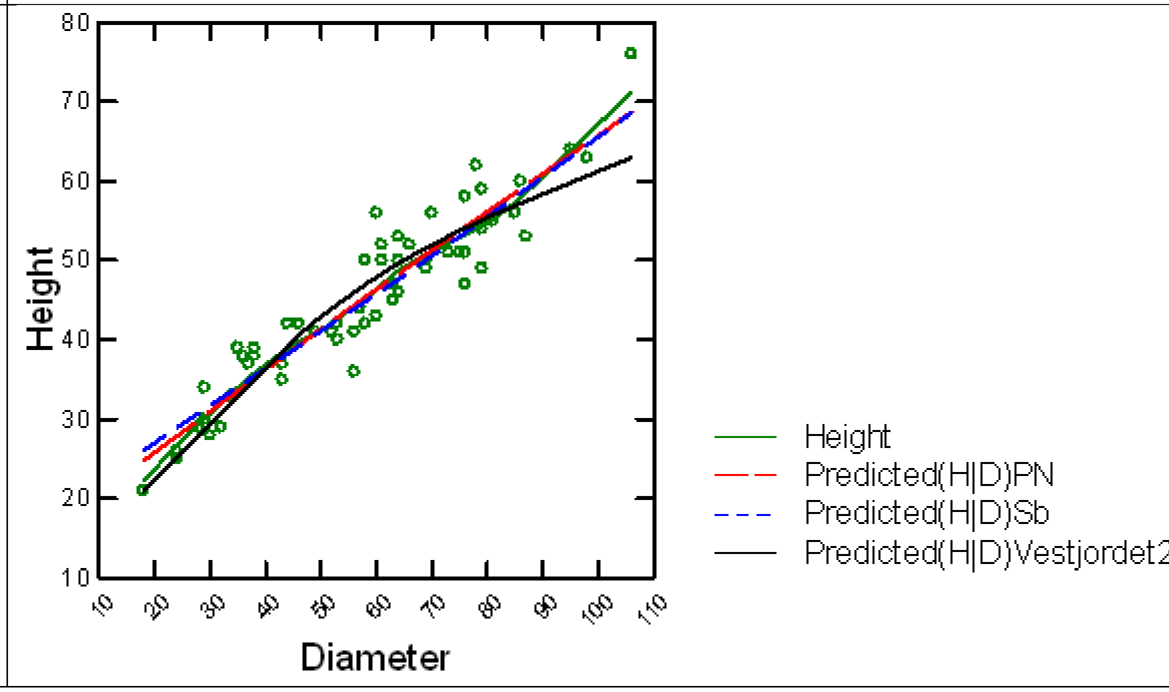




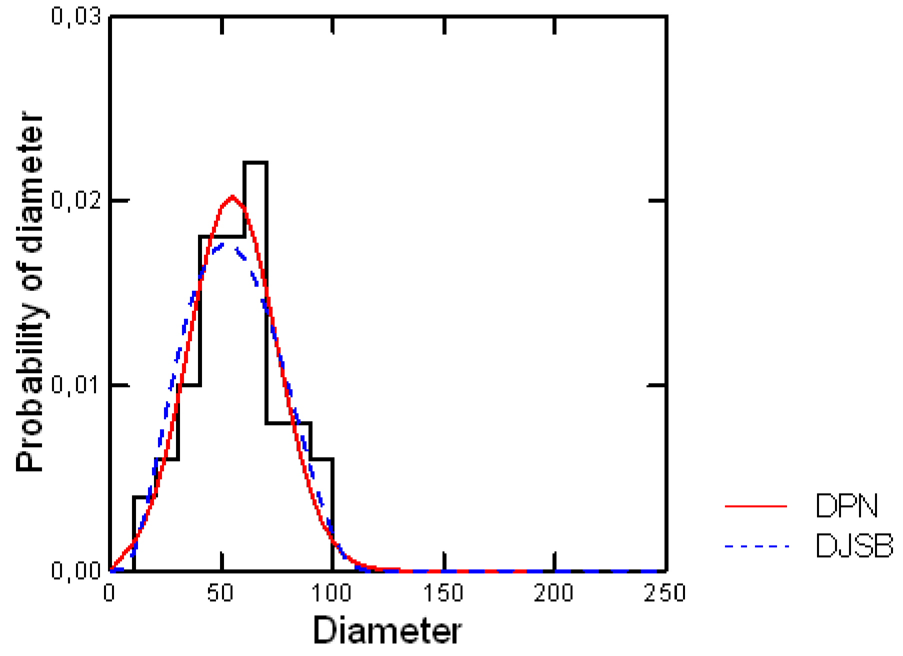
Results (↓) for STAND = 10

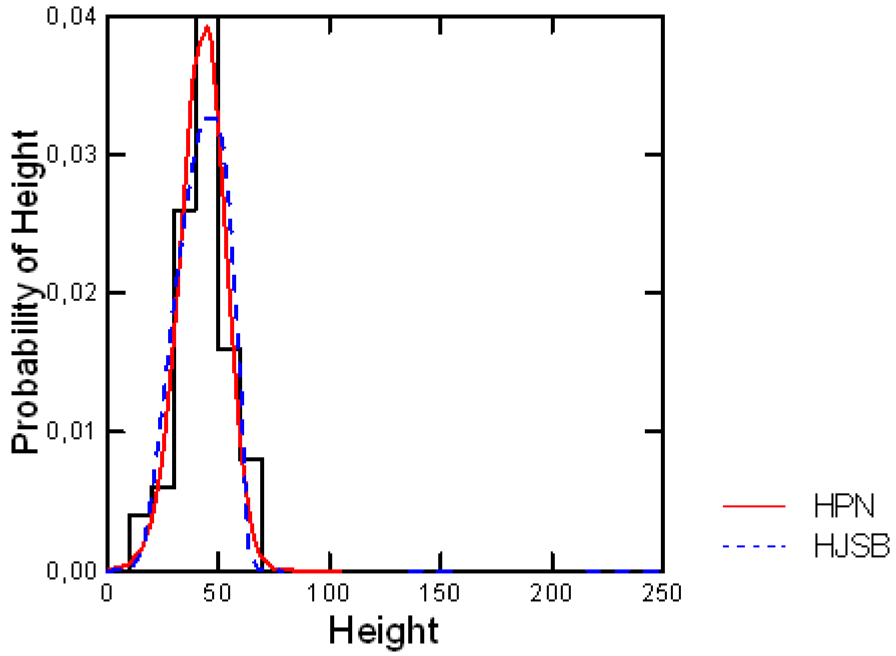


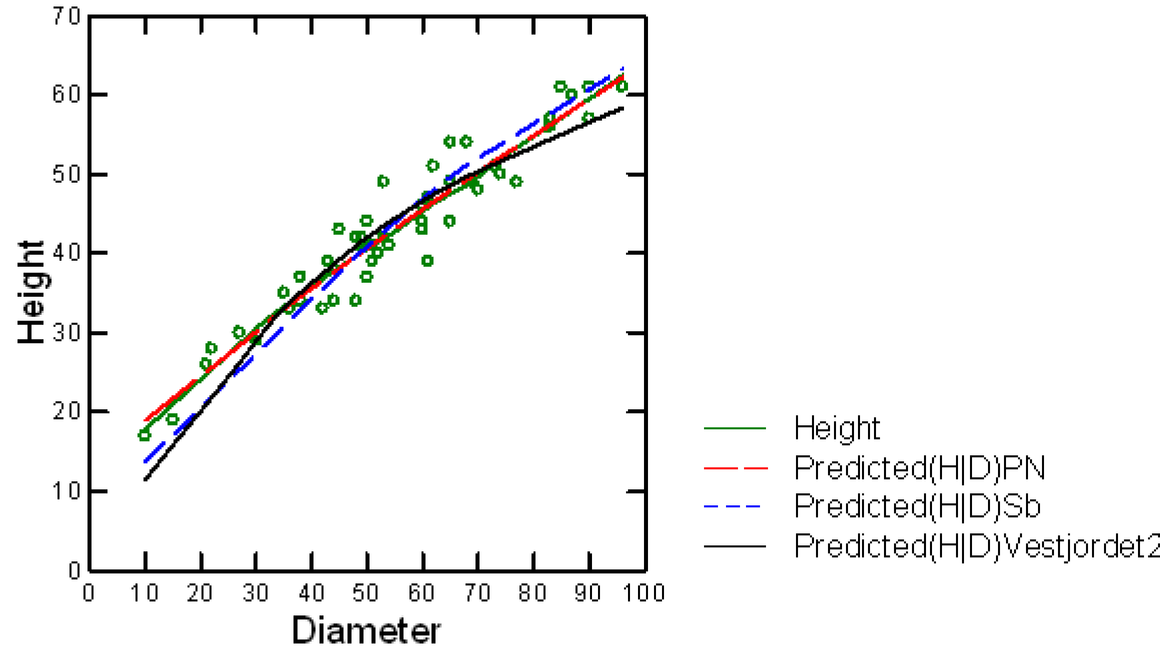




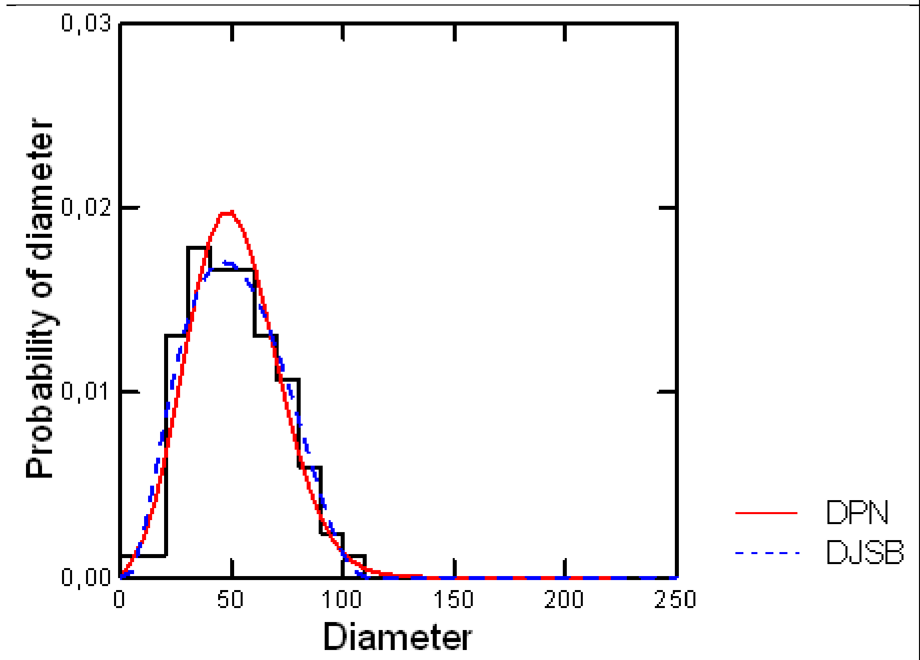
Results (↓) for STAND = 11

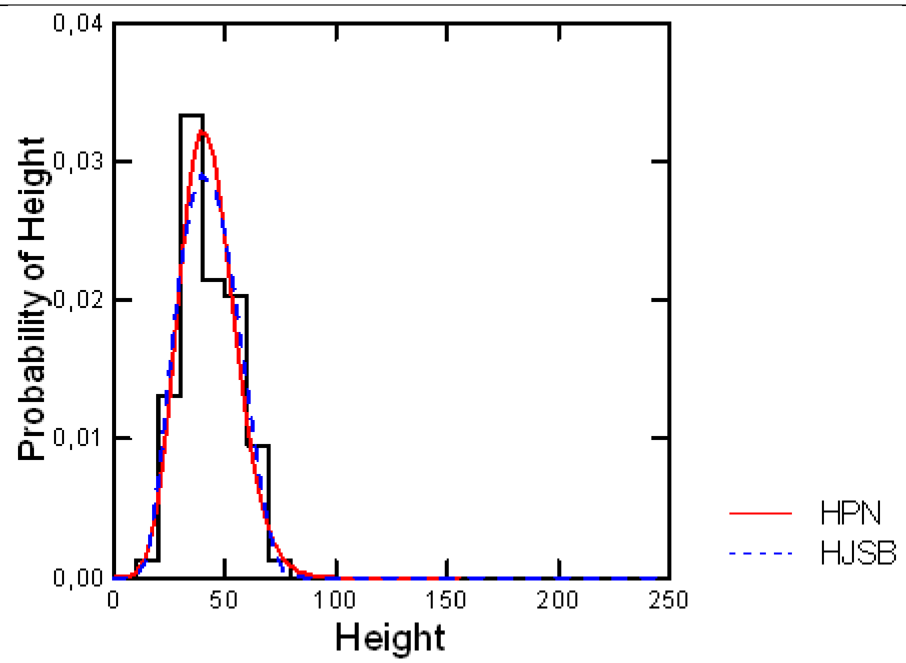


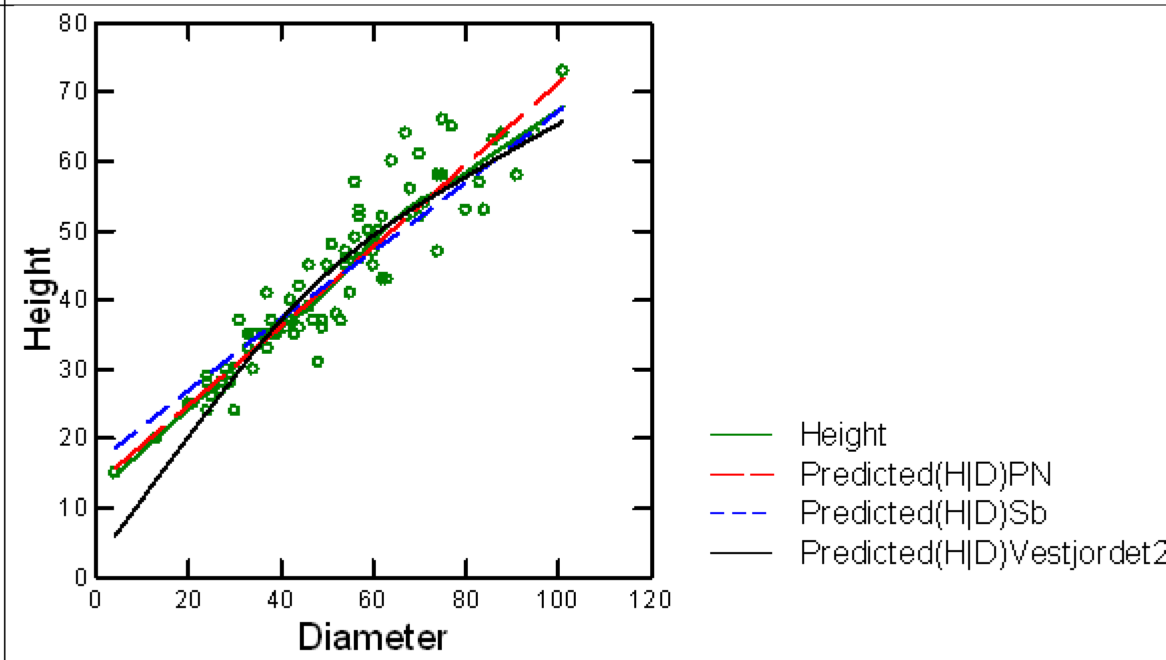




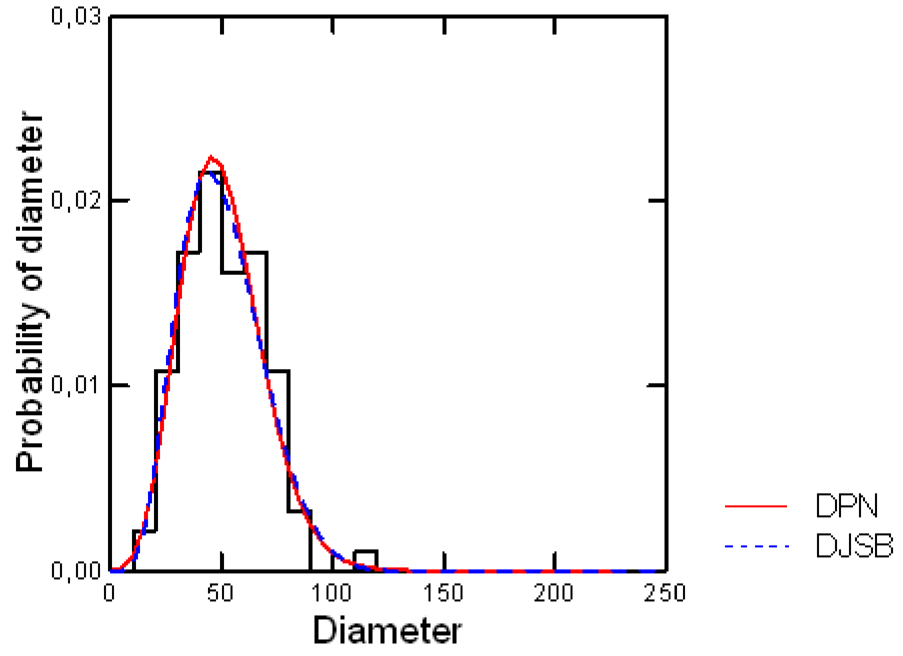
Results (↓) for STAND = 12

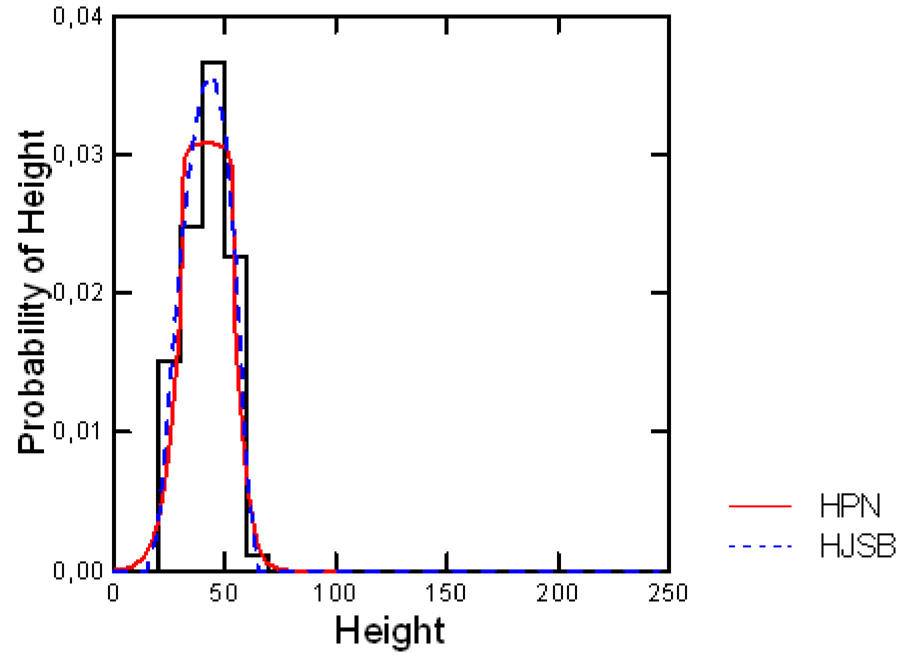


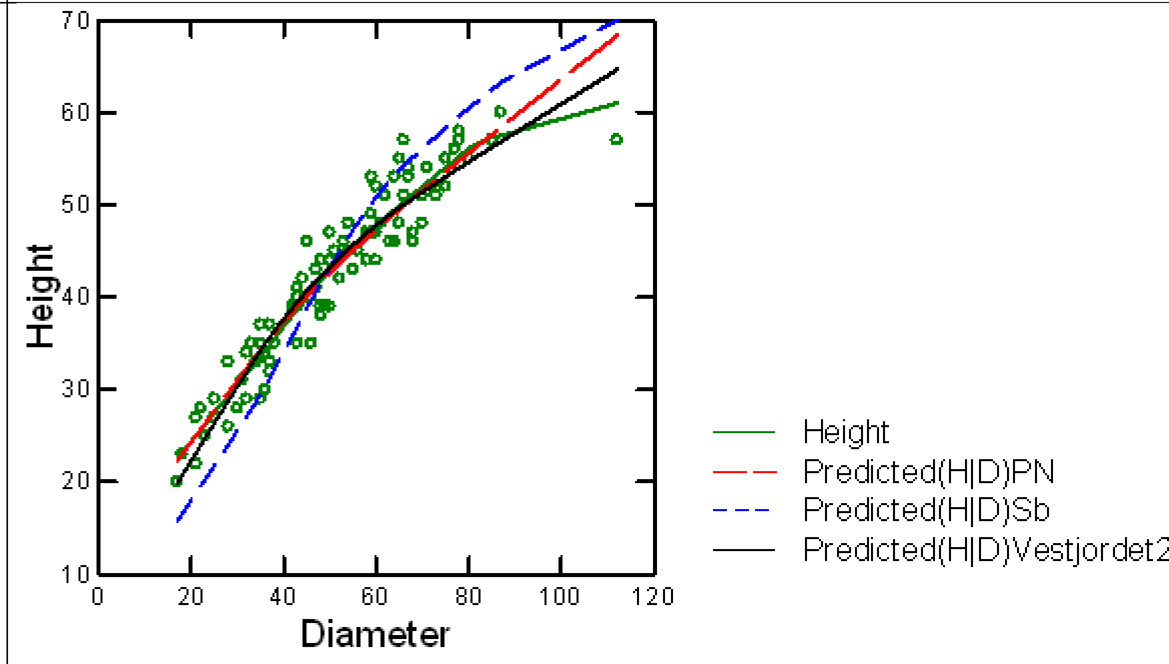




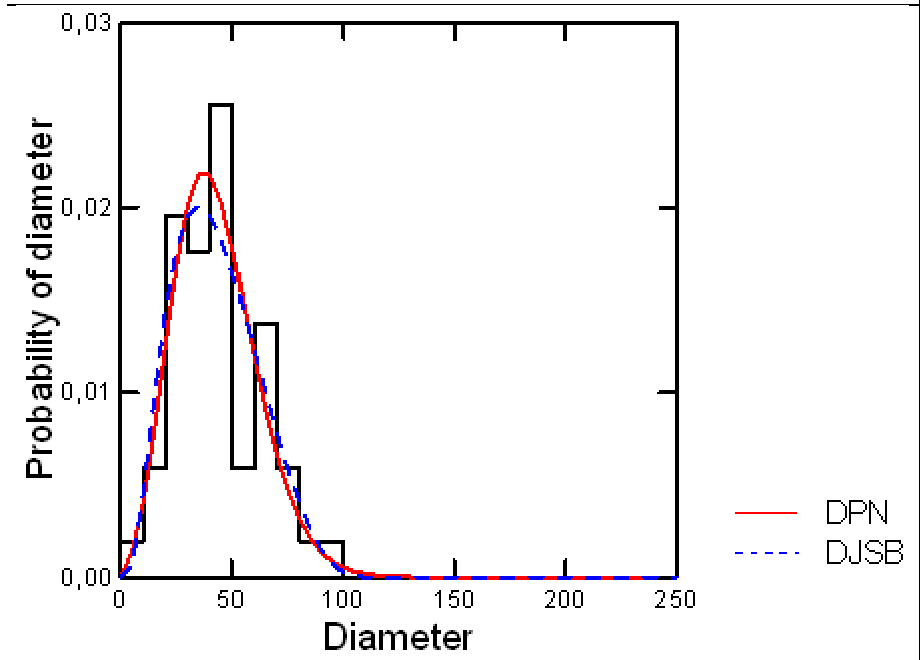
Results (↓) for STAND = 13

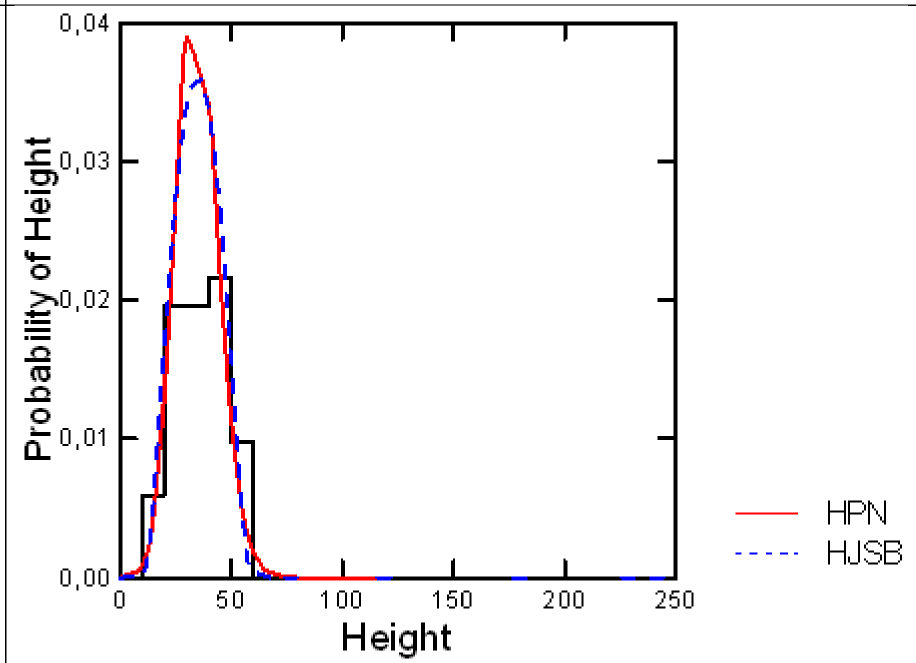


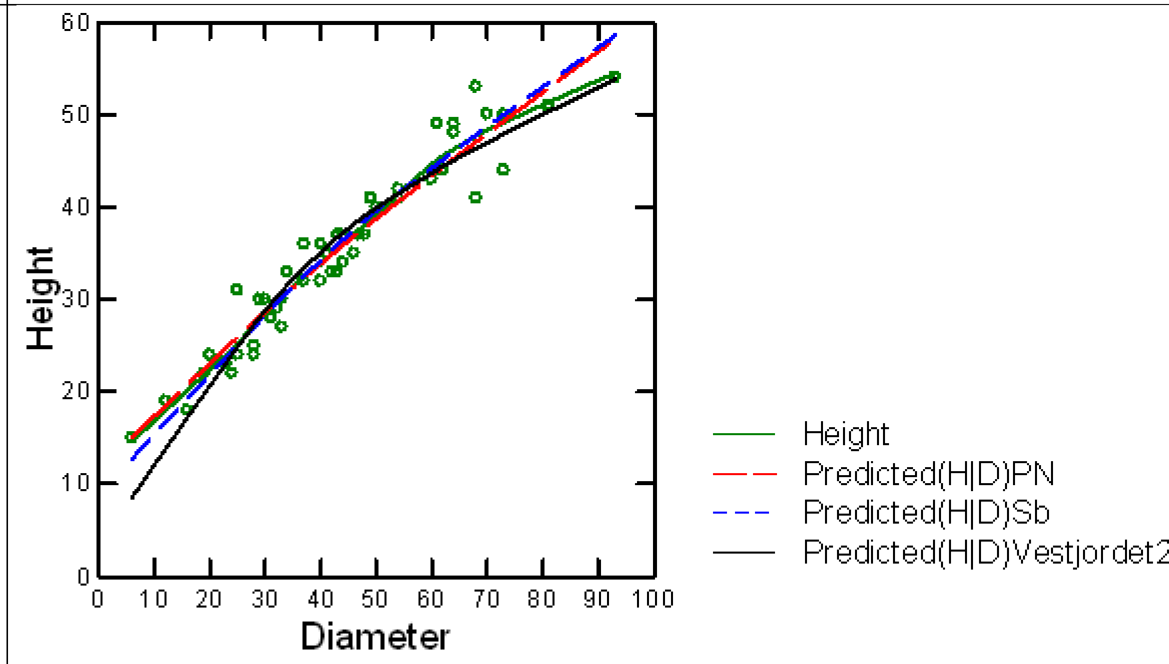




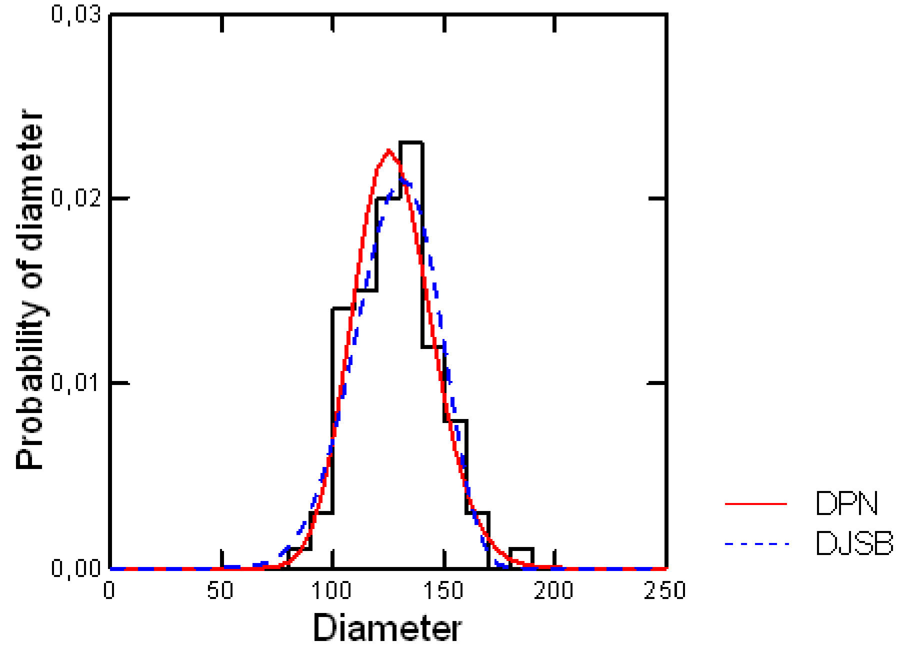
Results (↓) for STAND = 14

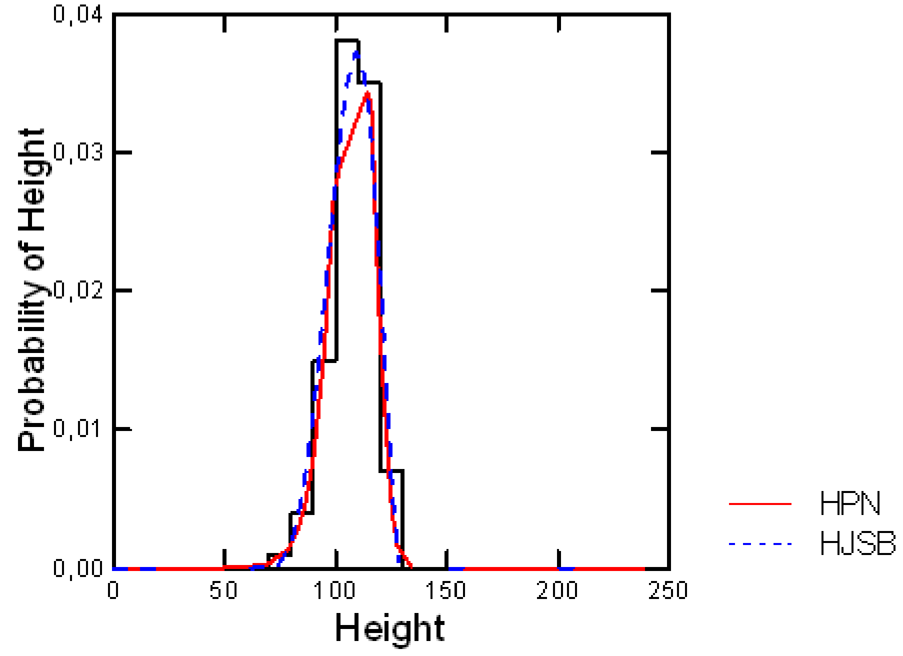


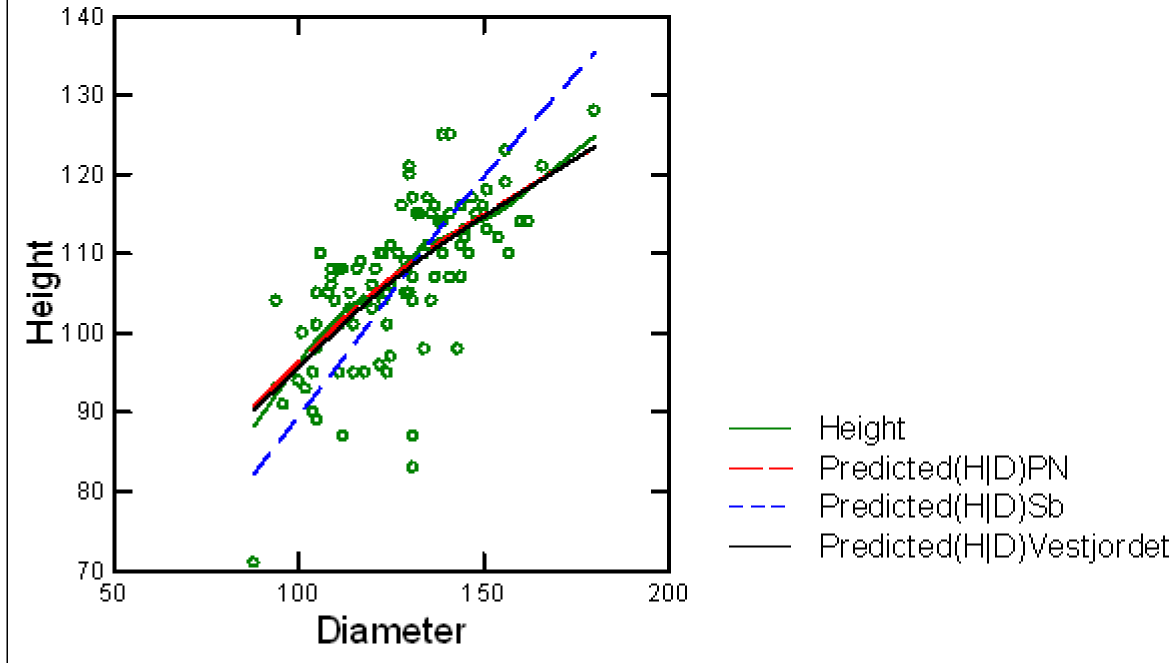




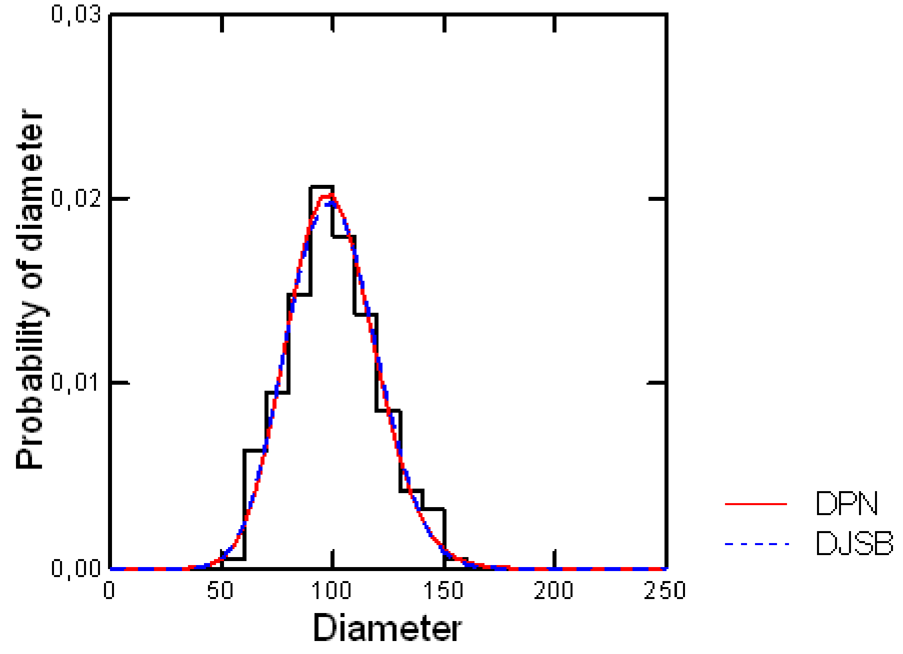
Results (↓) for STAND = 15

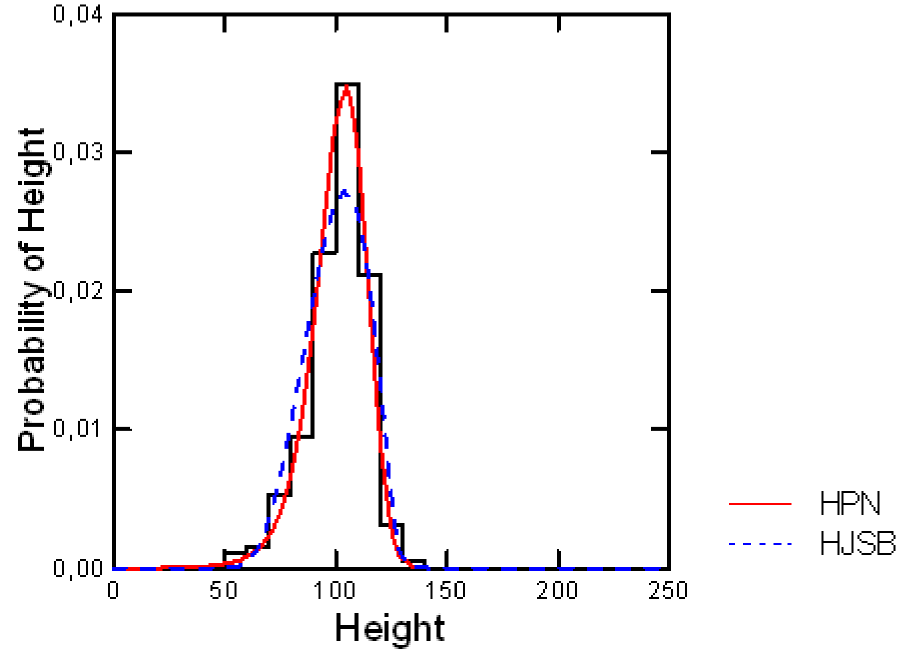


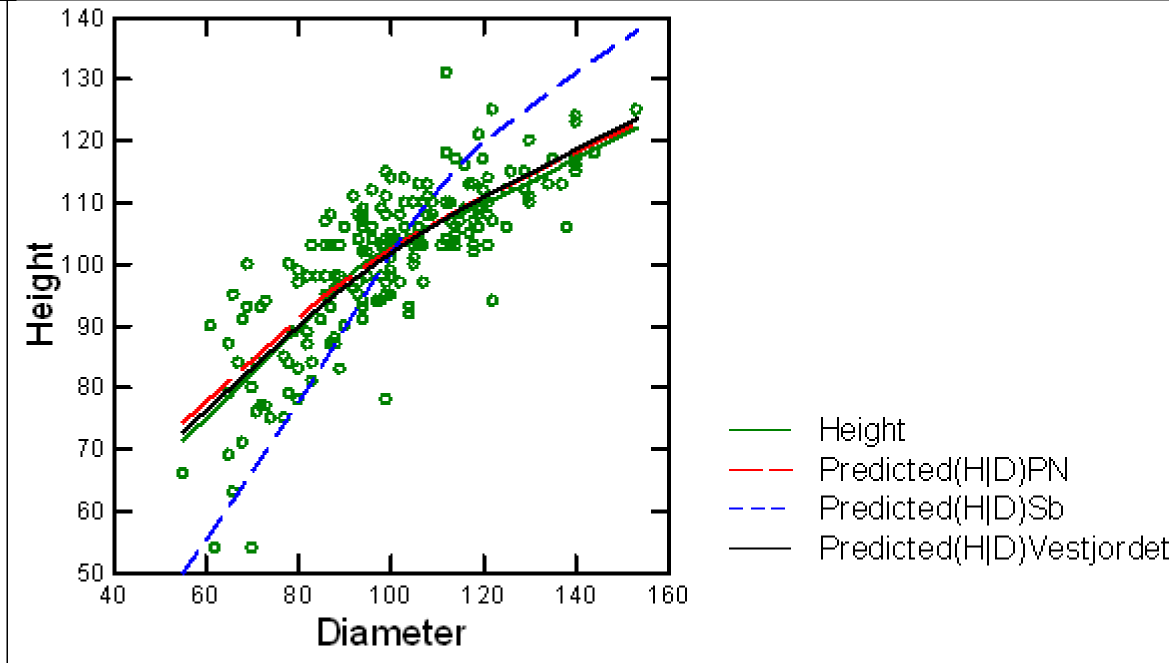




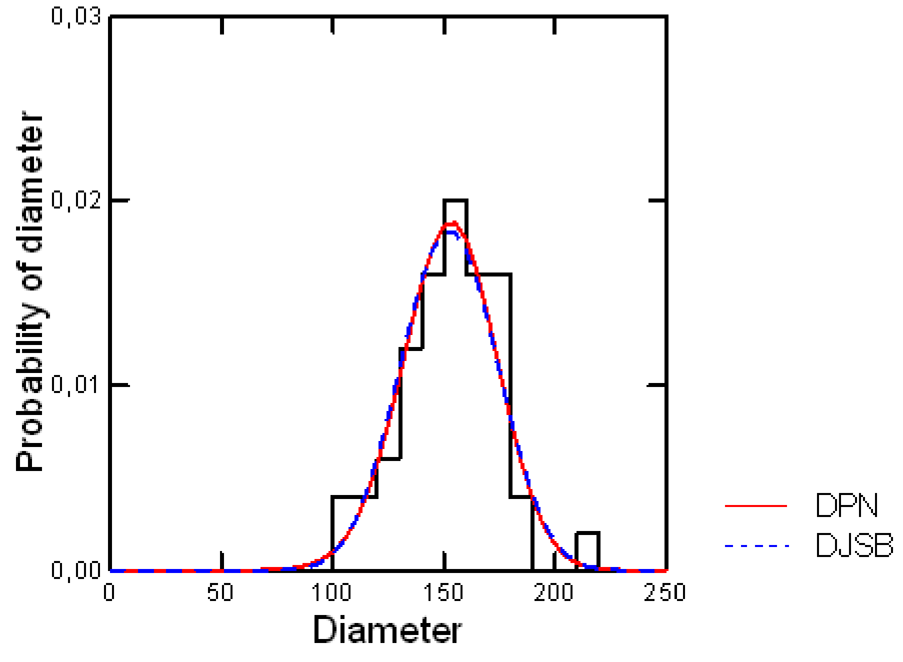
Results (↓) for STAND = 16

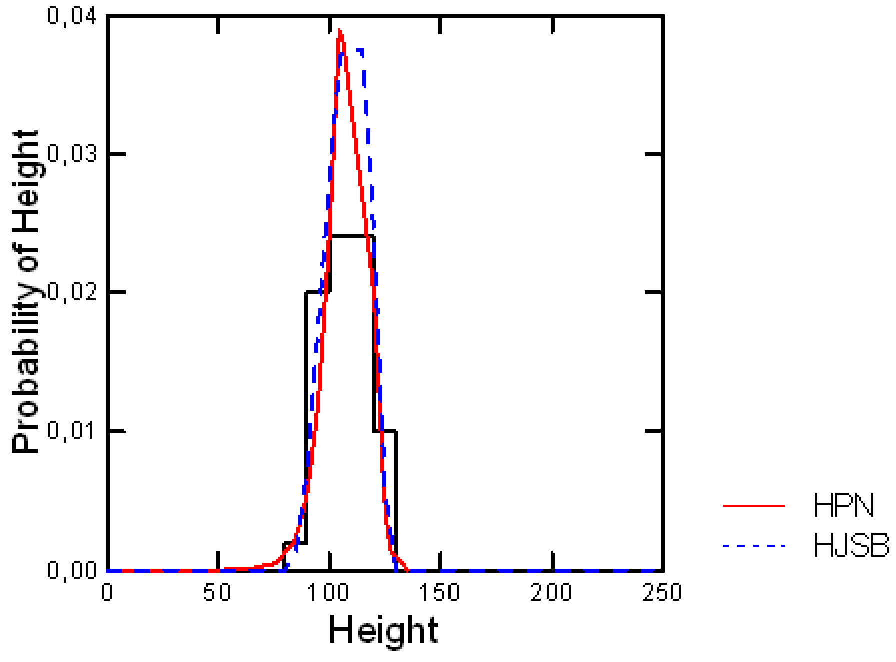


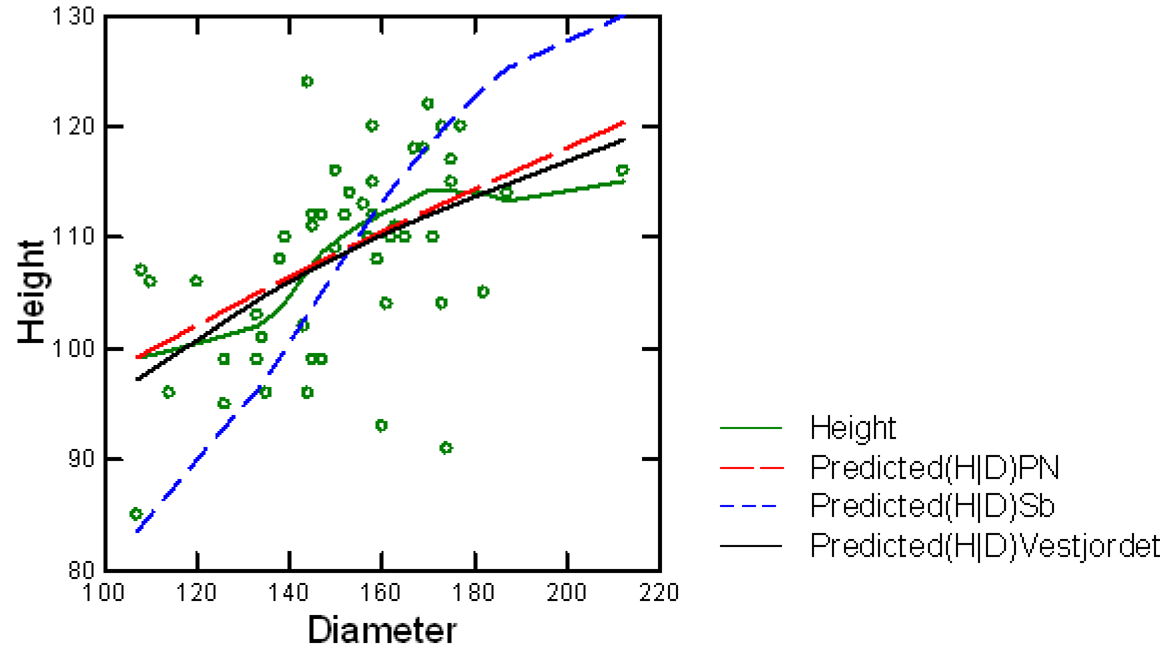




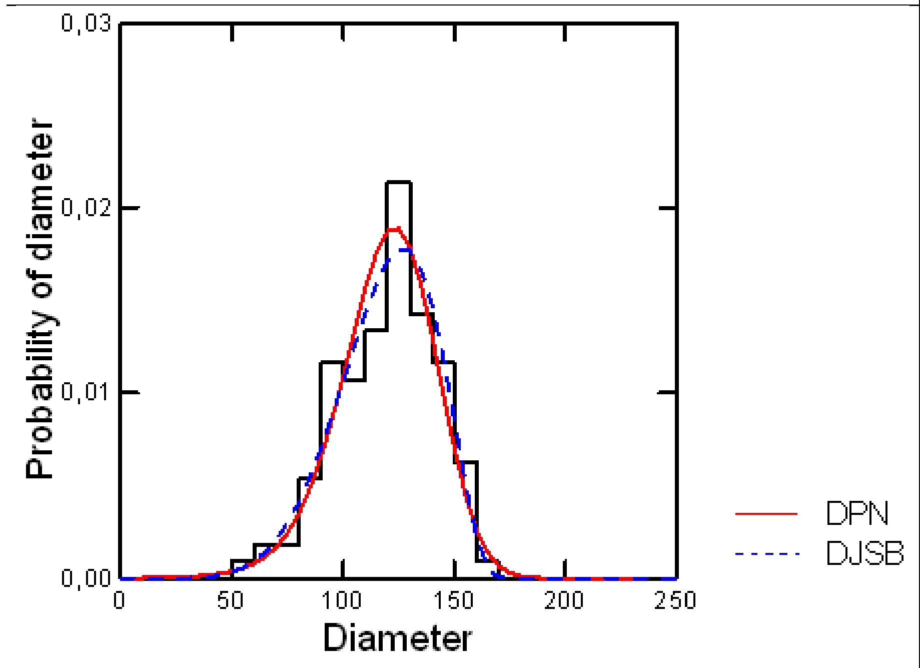
Results (↓) for STAND = 17

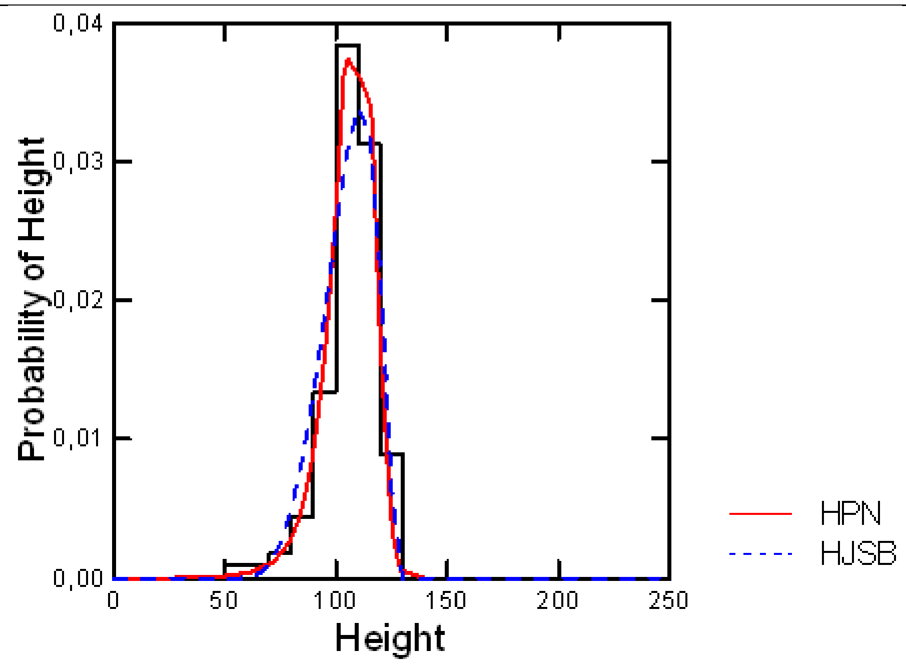


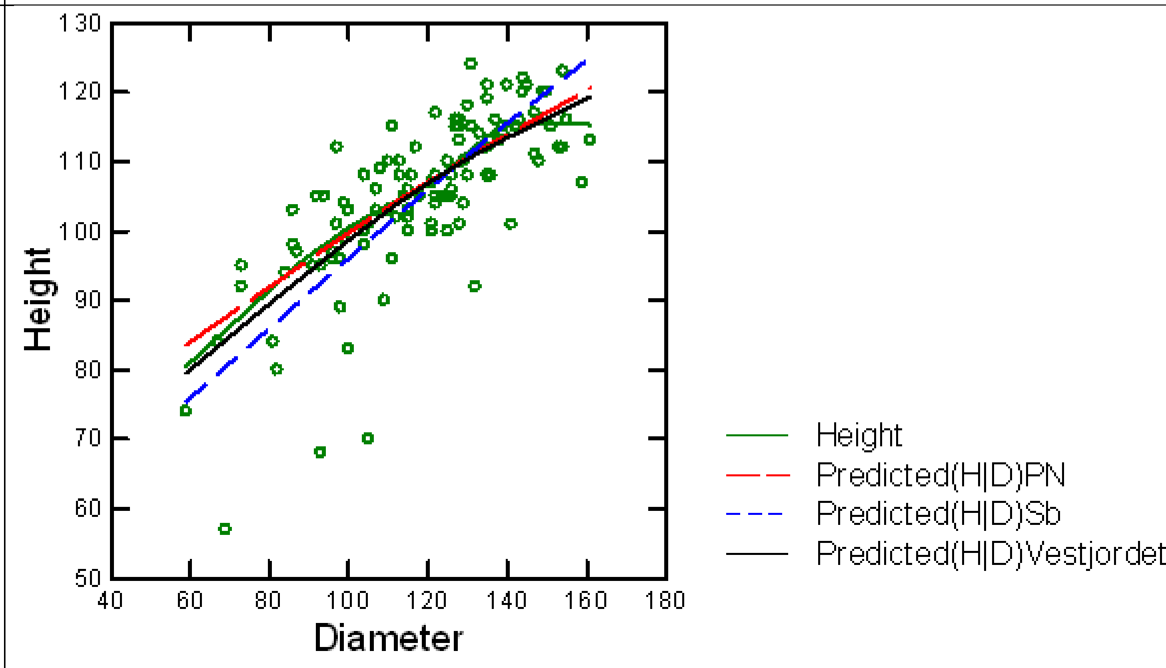




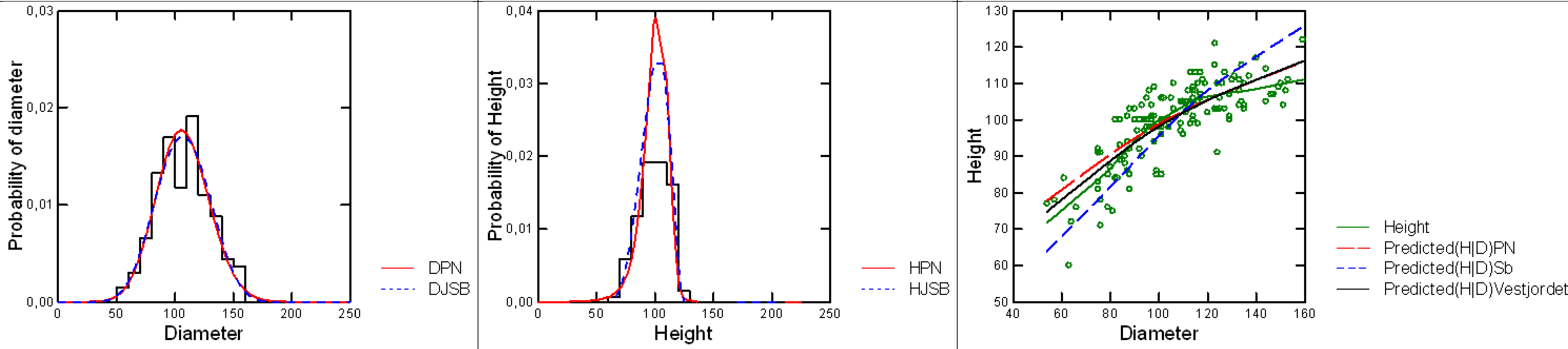
Results (↓) for STAND = 18



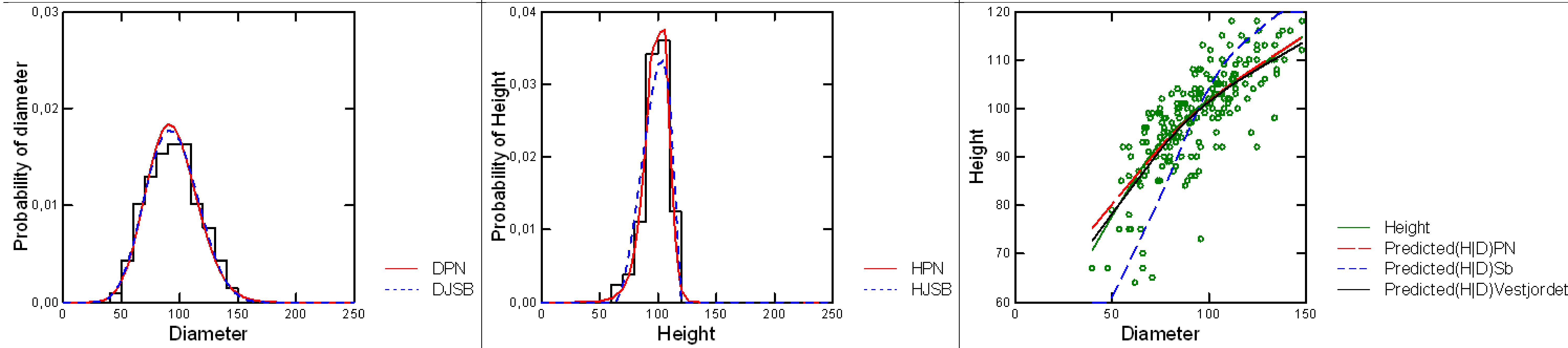




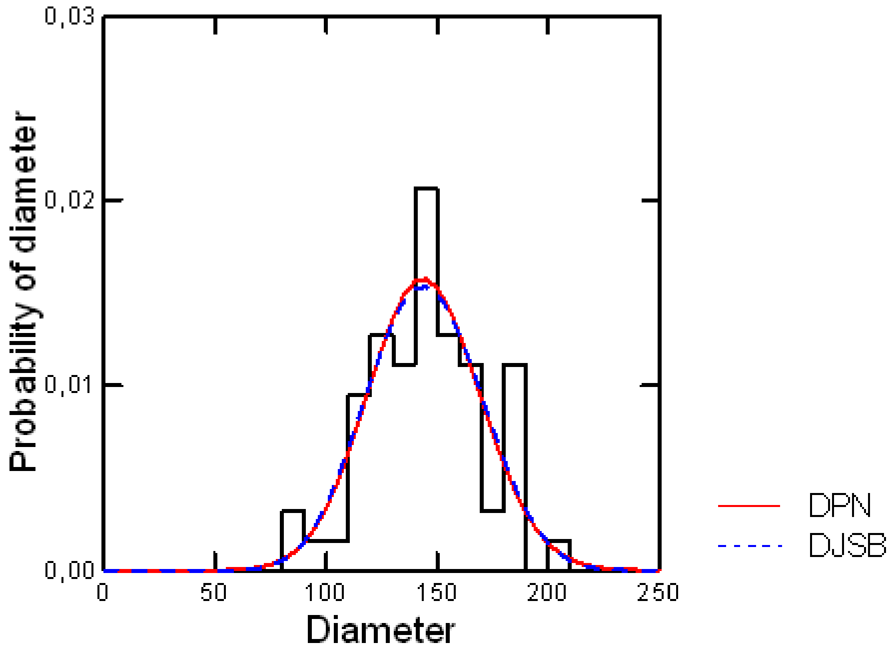
Results (↓) for STAND = 19

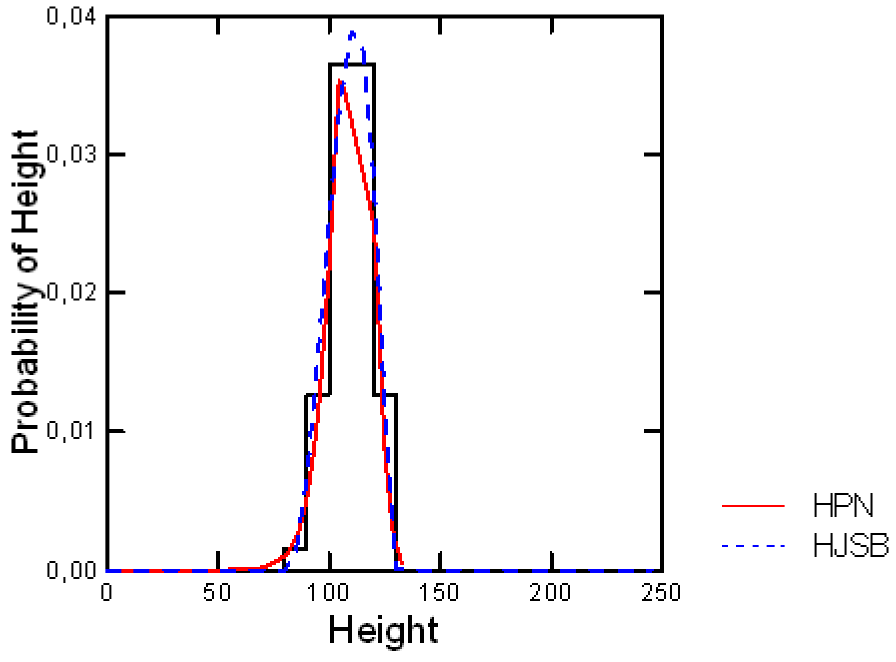


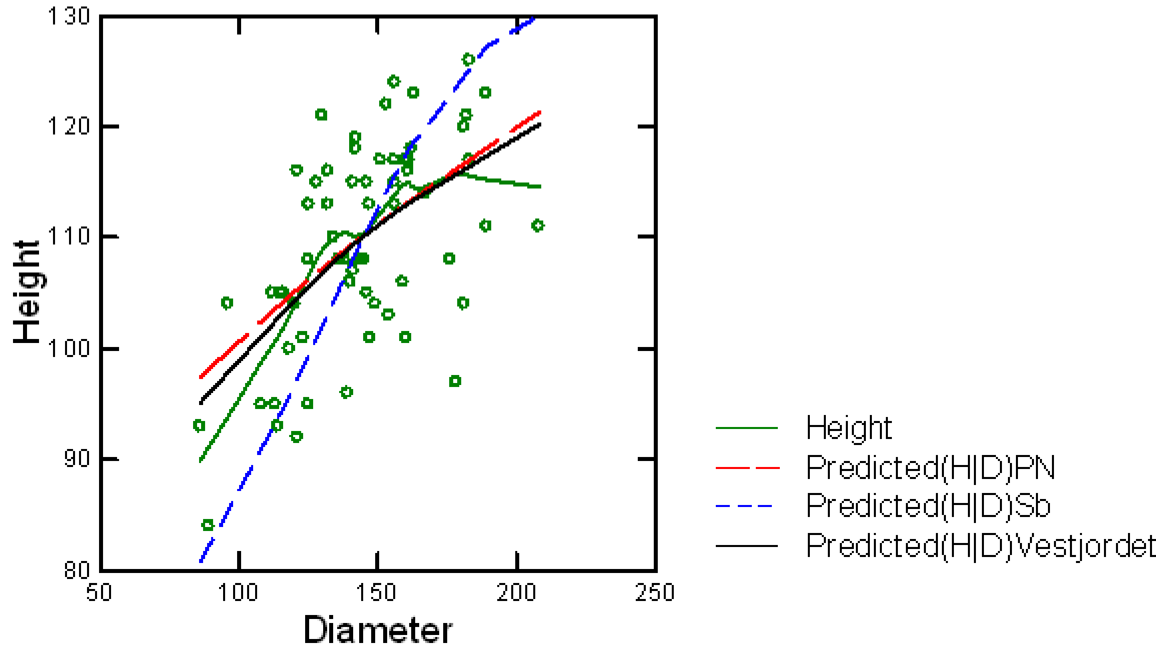
Results (↓) for STAND = 20



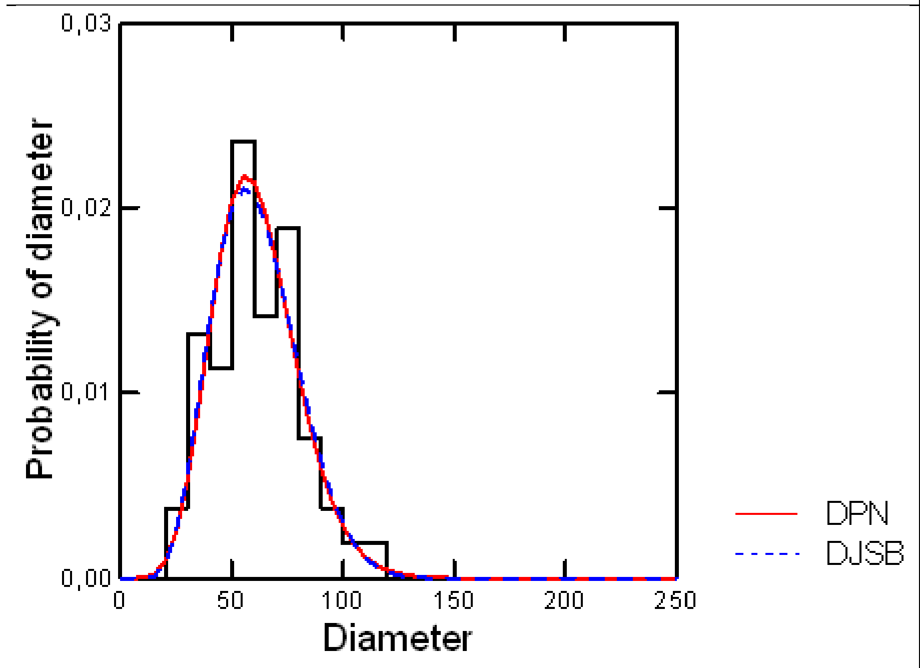
Results (↓) for STAND = 21

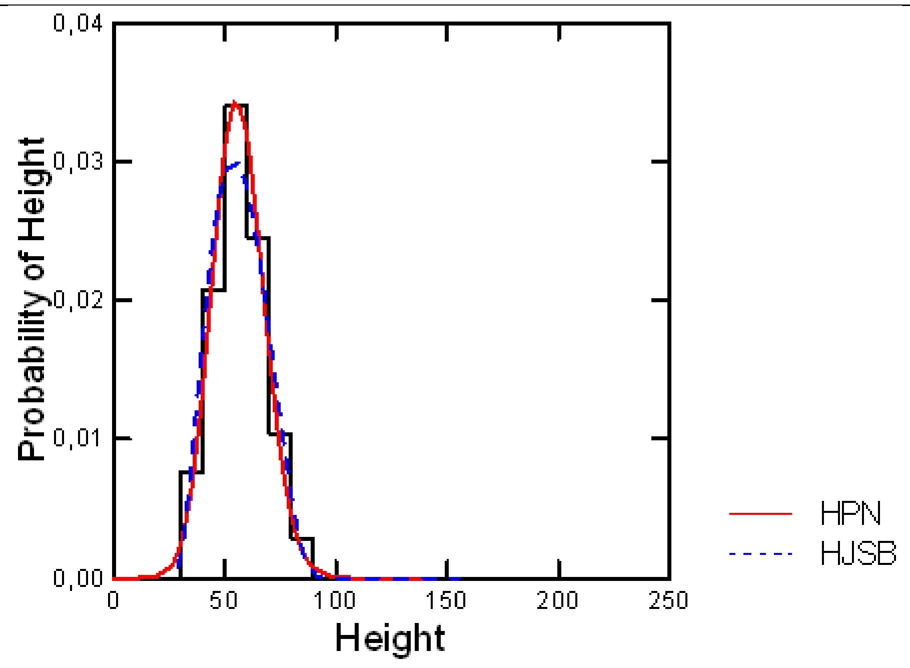


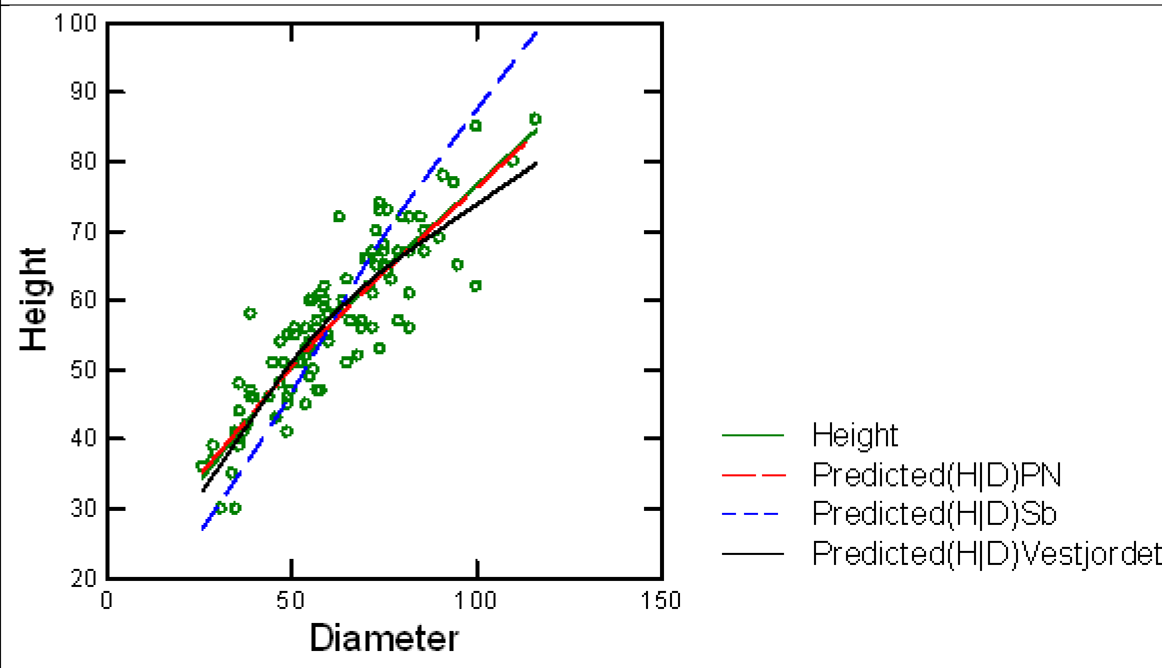




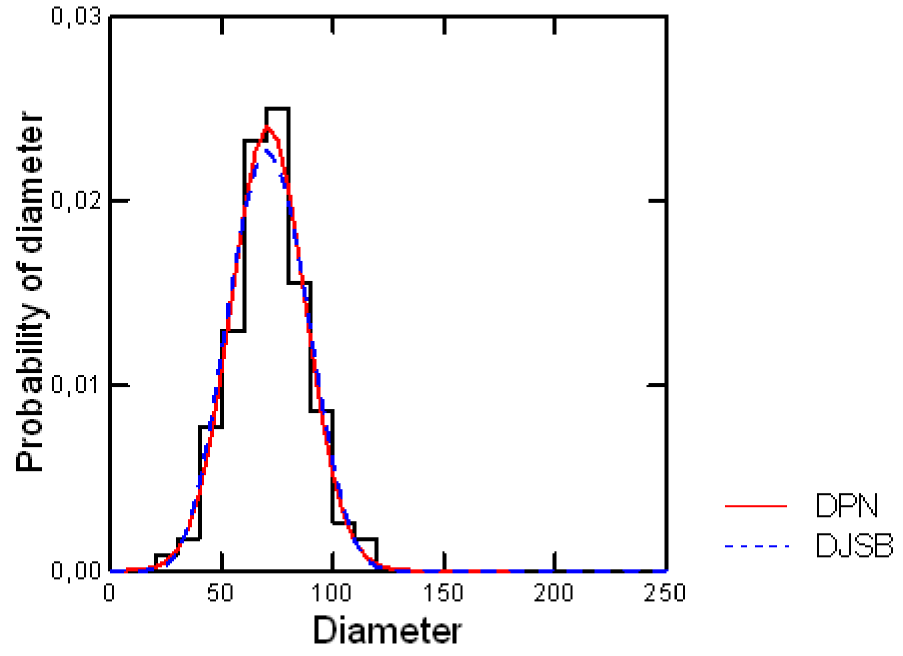
Results (↓) for STAND = 22

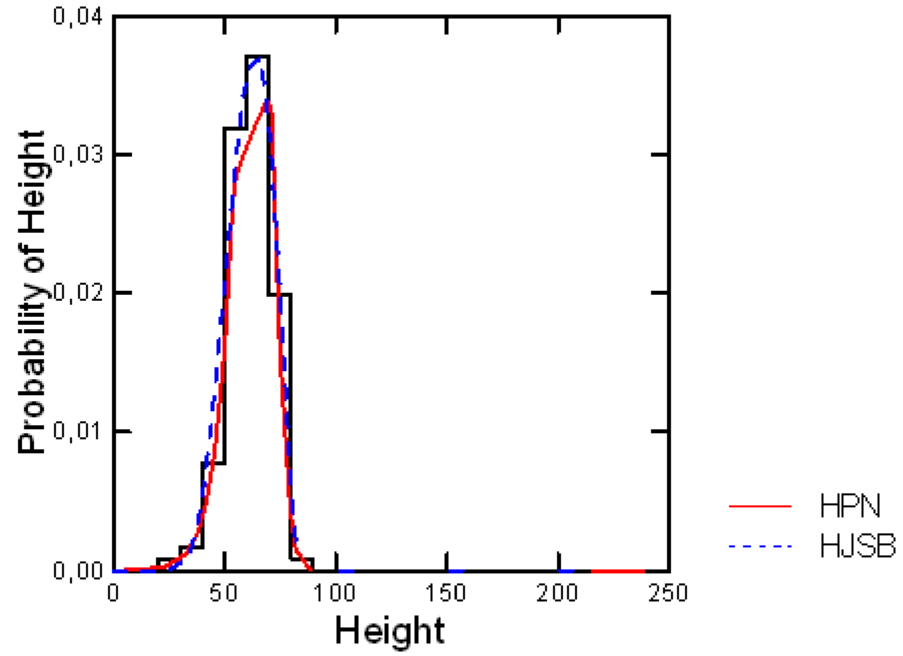


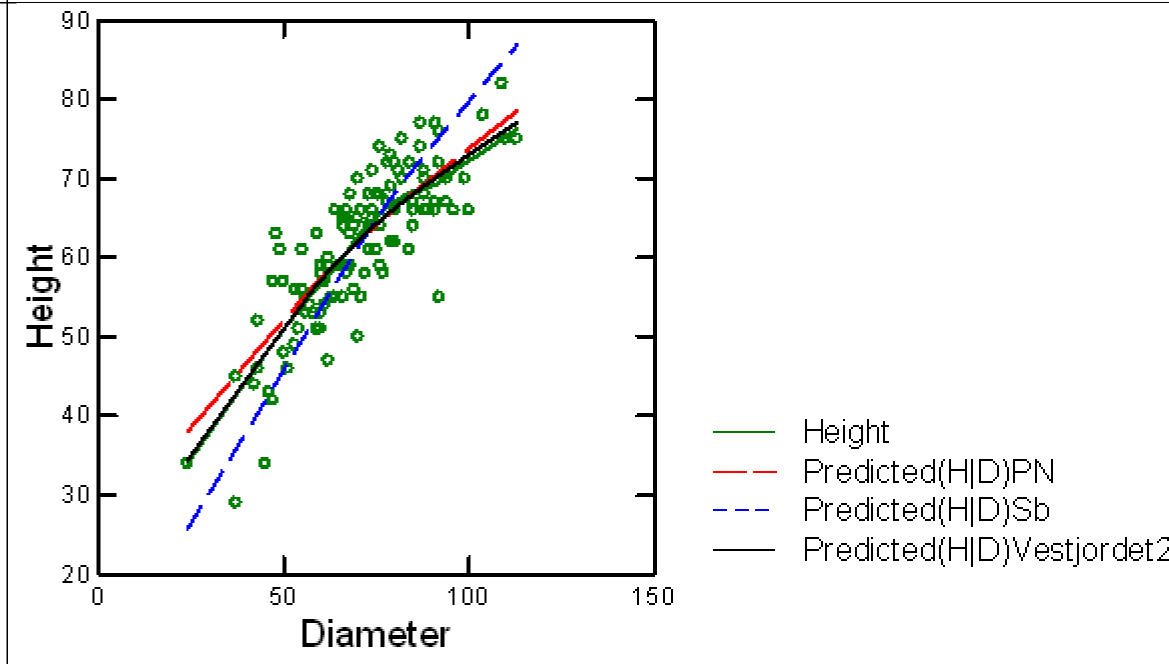




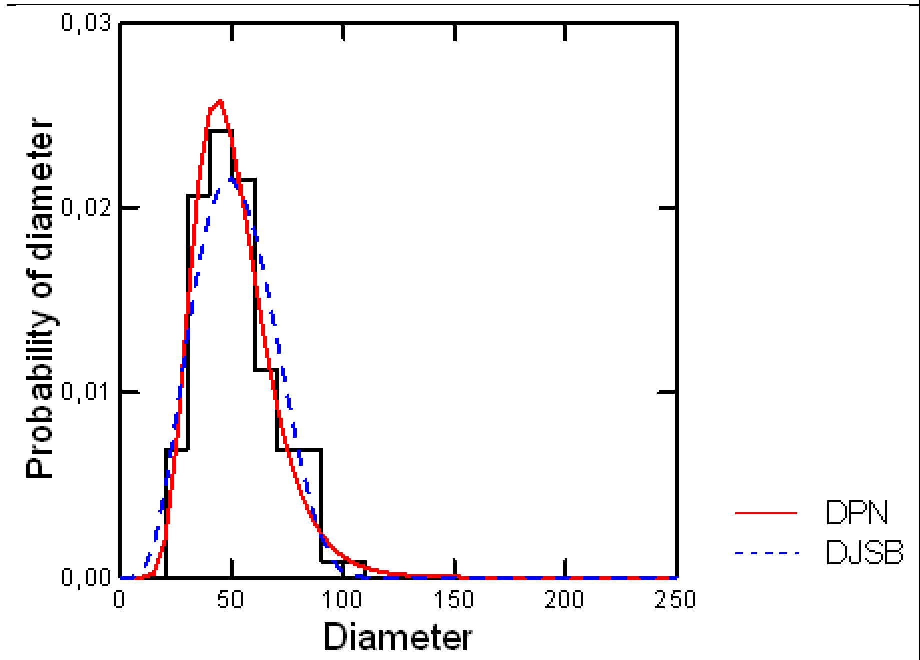
Results (↓) for STAND = 23

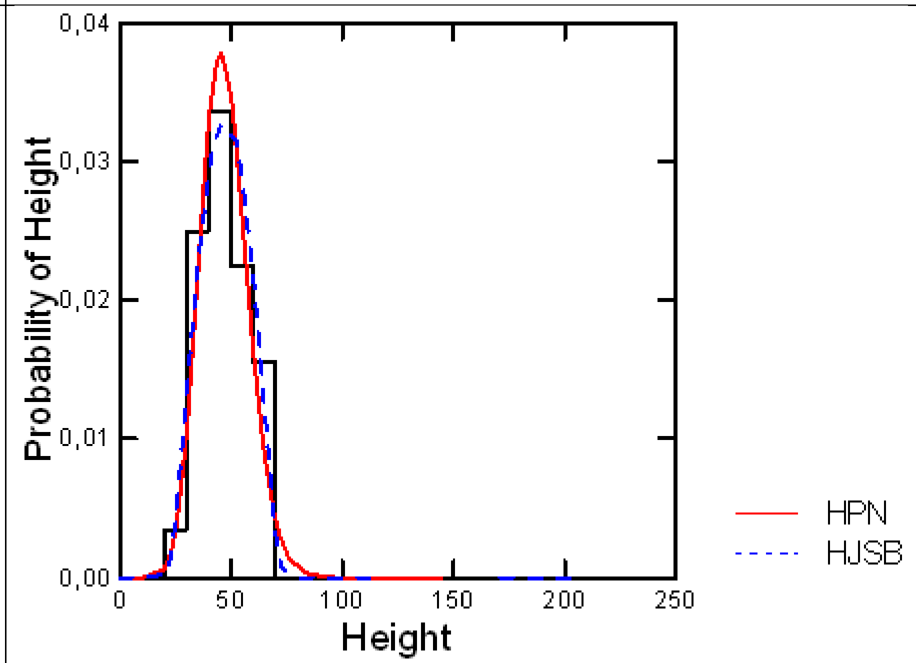


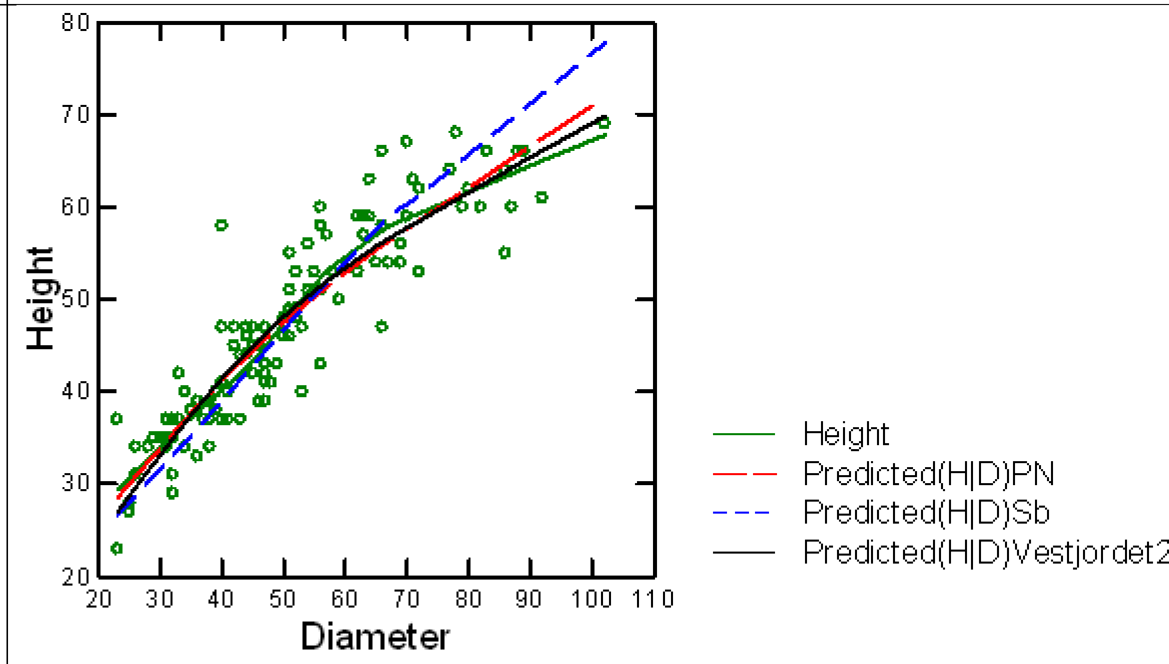




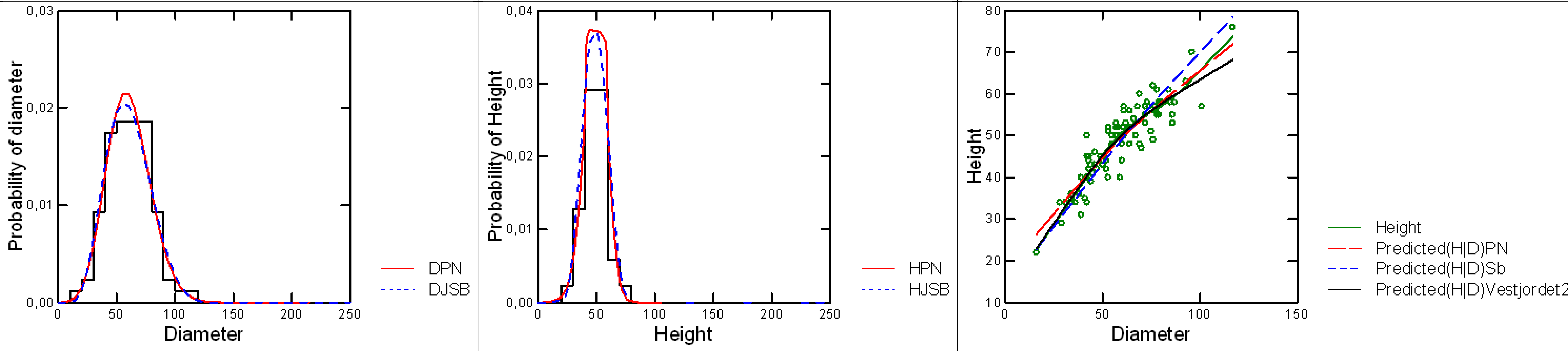
Results (↓) for STAND = 24



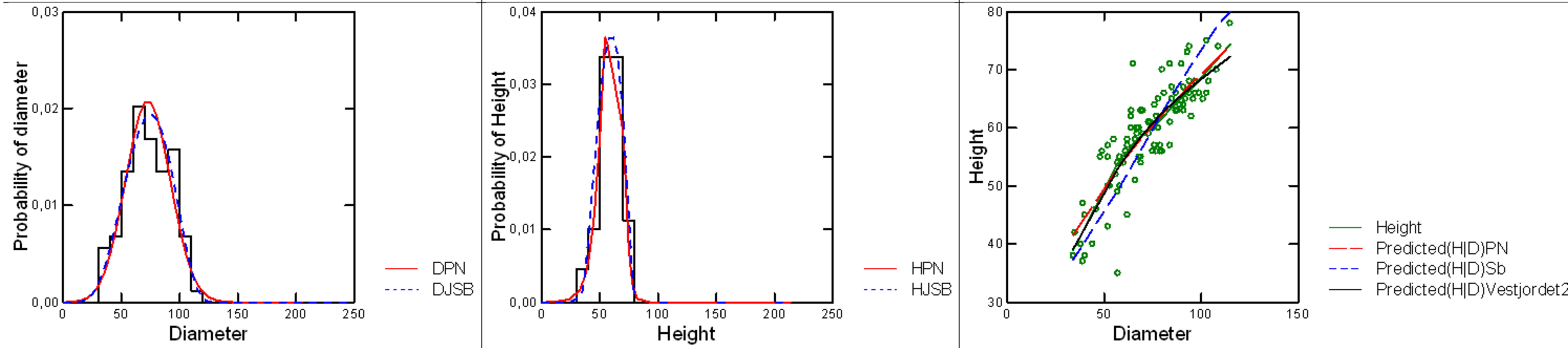




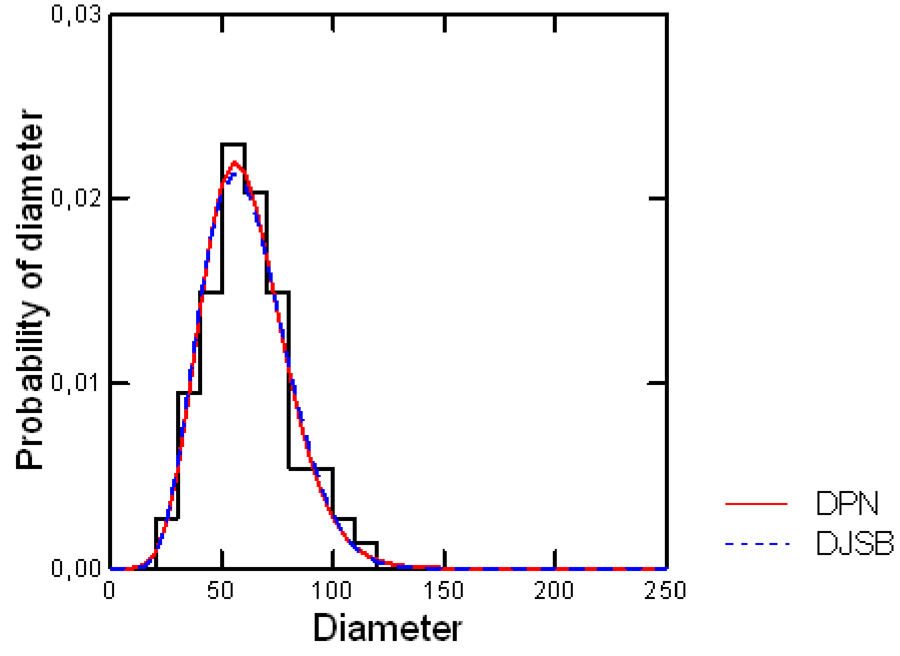
Results (↓) for STAND = 25

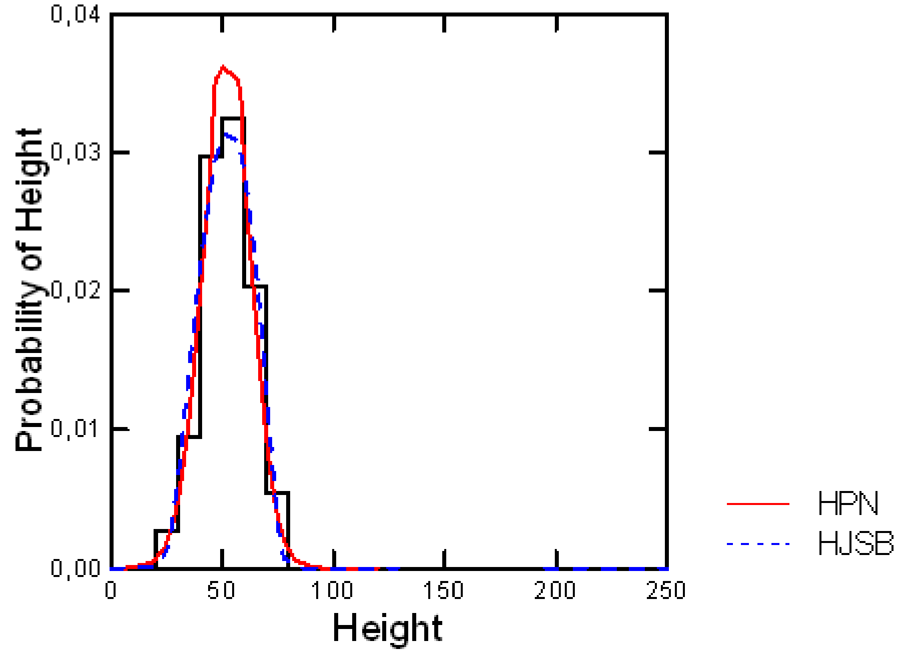


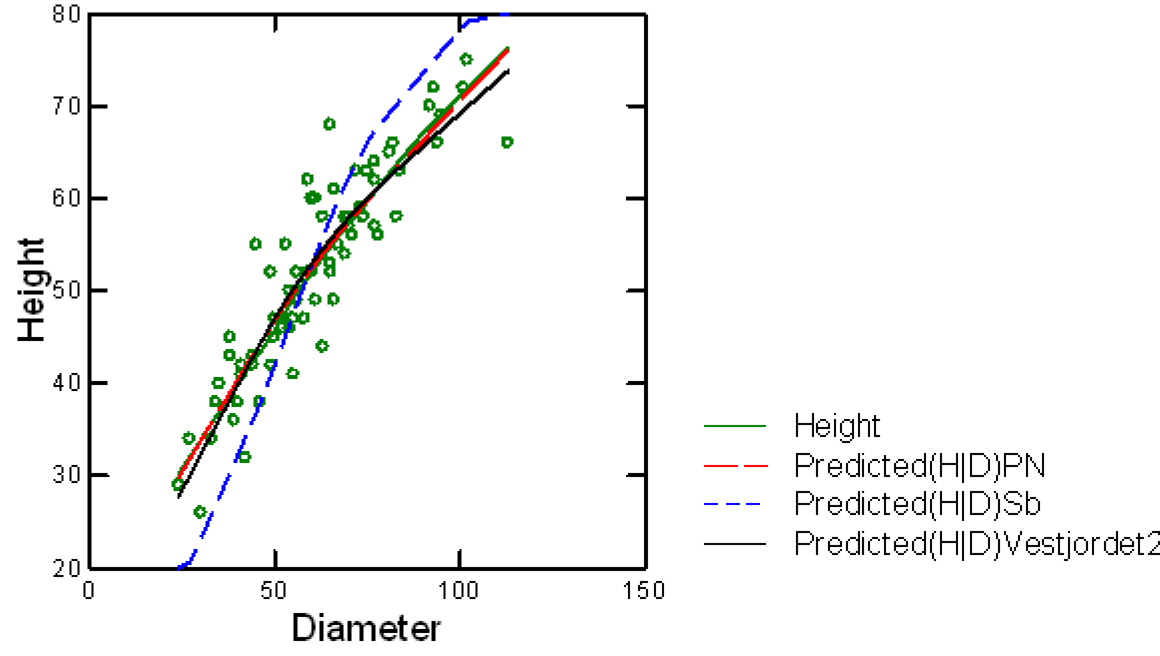
Results (↓) for STAND = 26



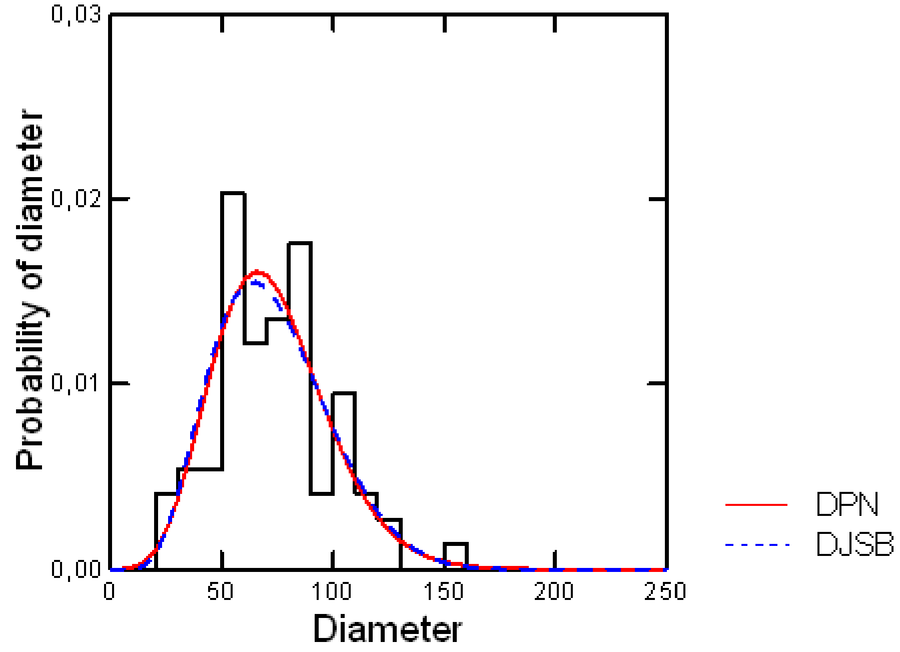
Results (↓) for STAND = 27

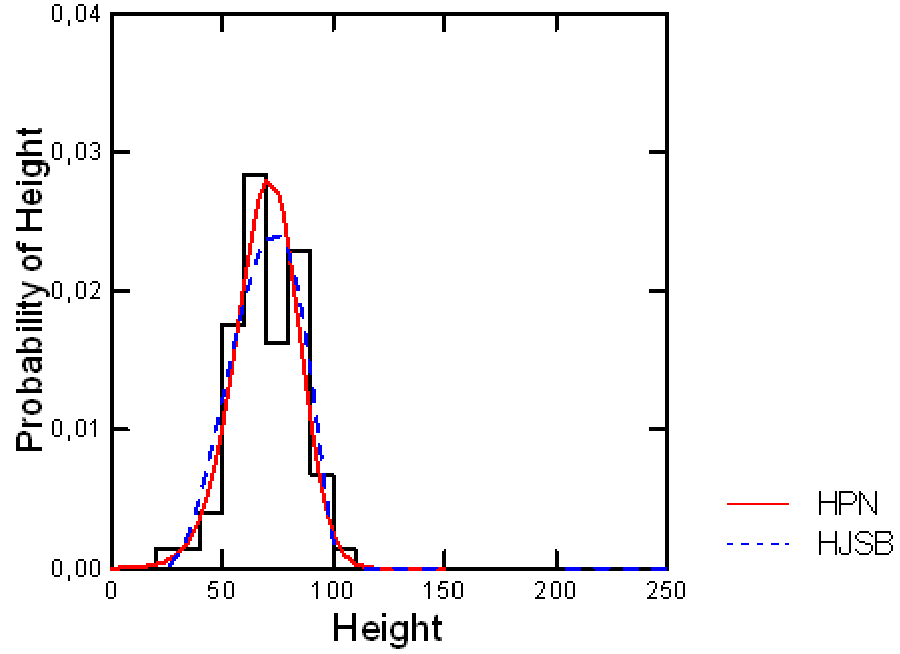


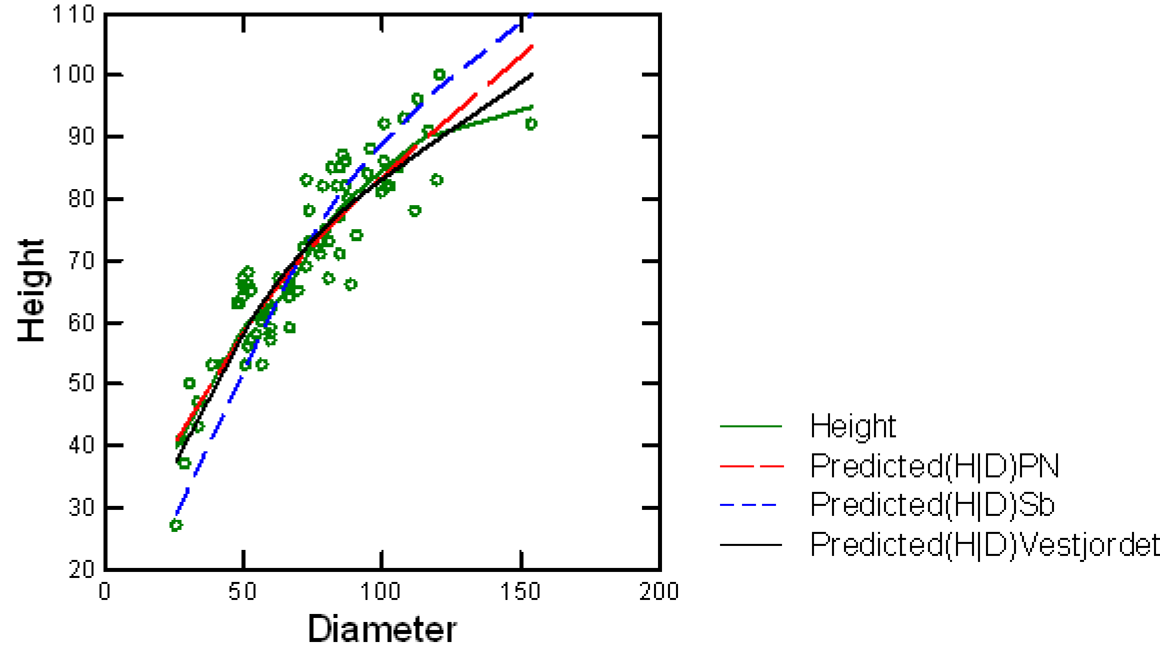




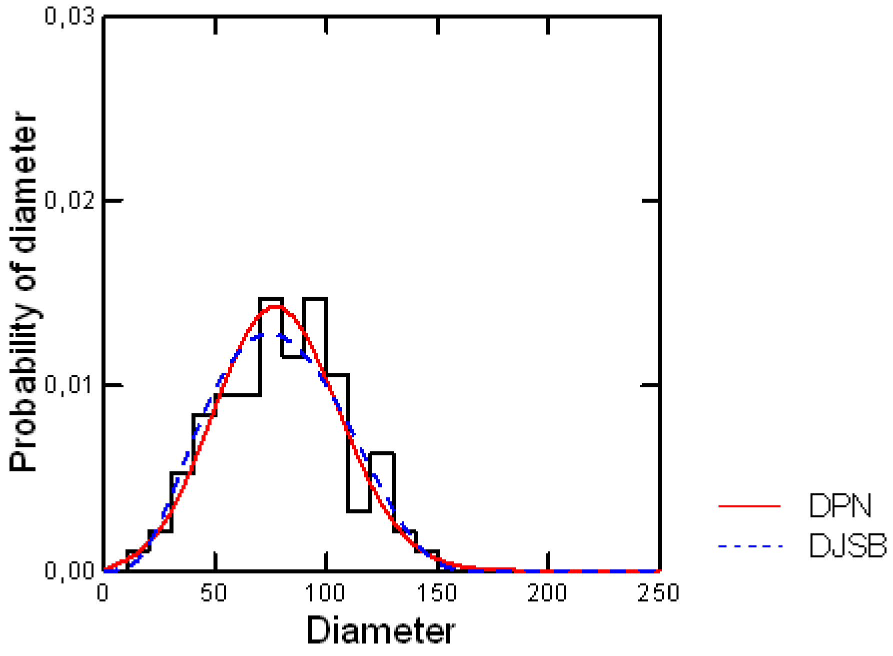
Results (↓) for STAND = 28

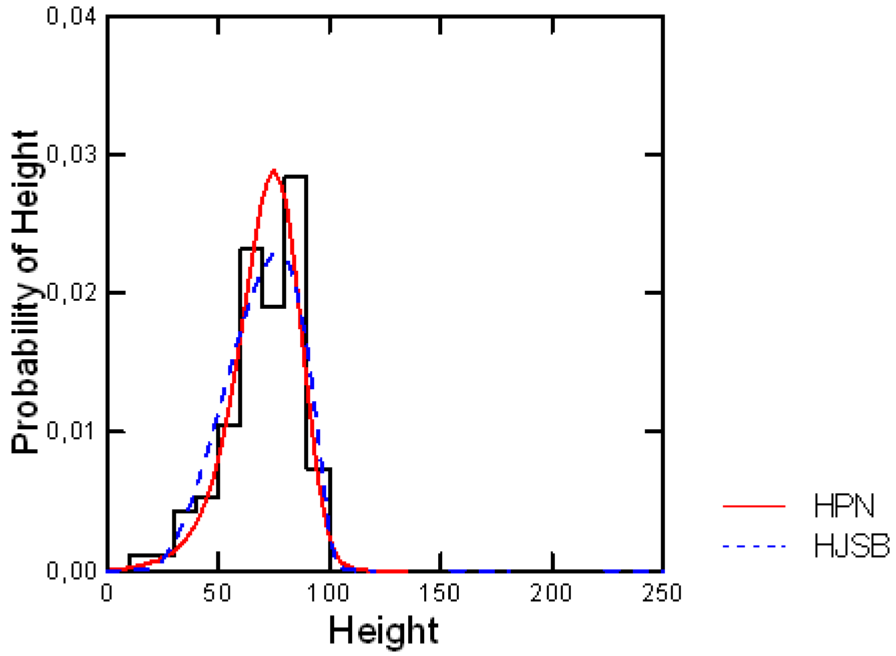


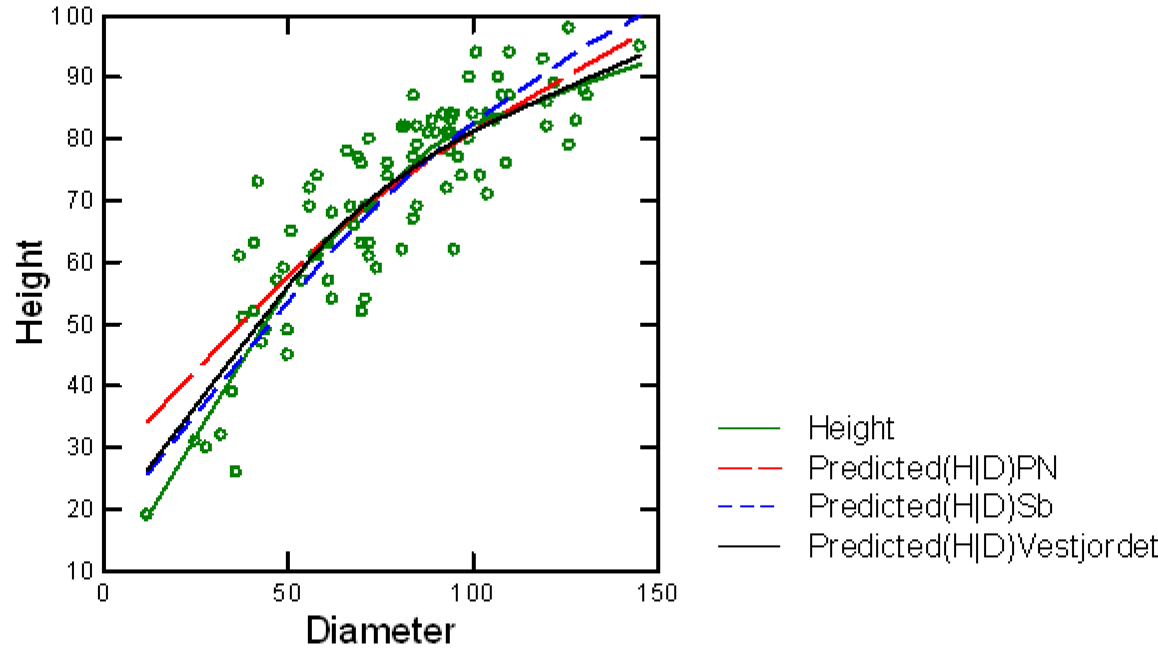




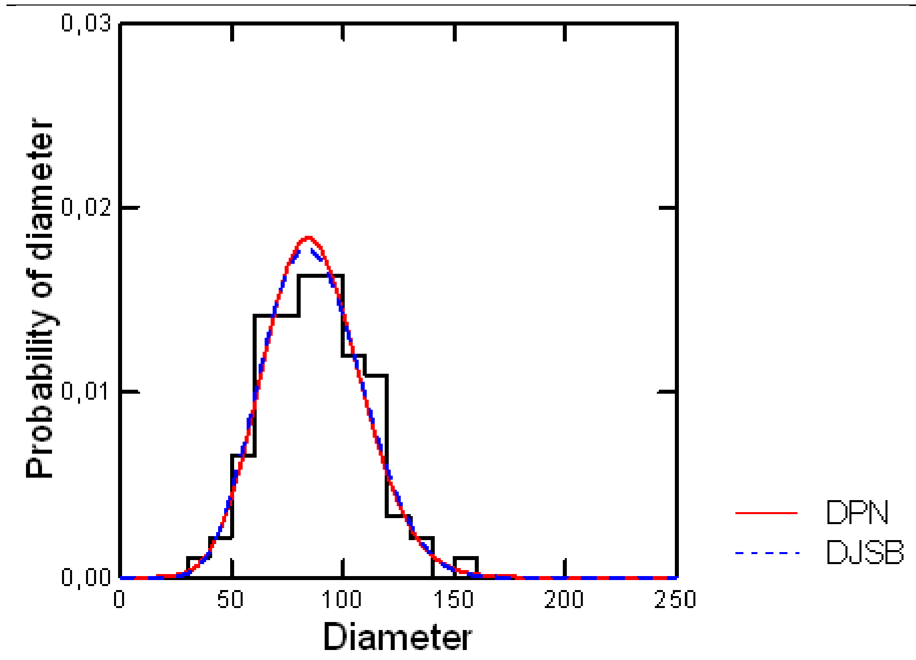
Results (↓) for STAND = 29

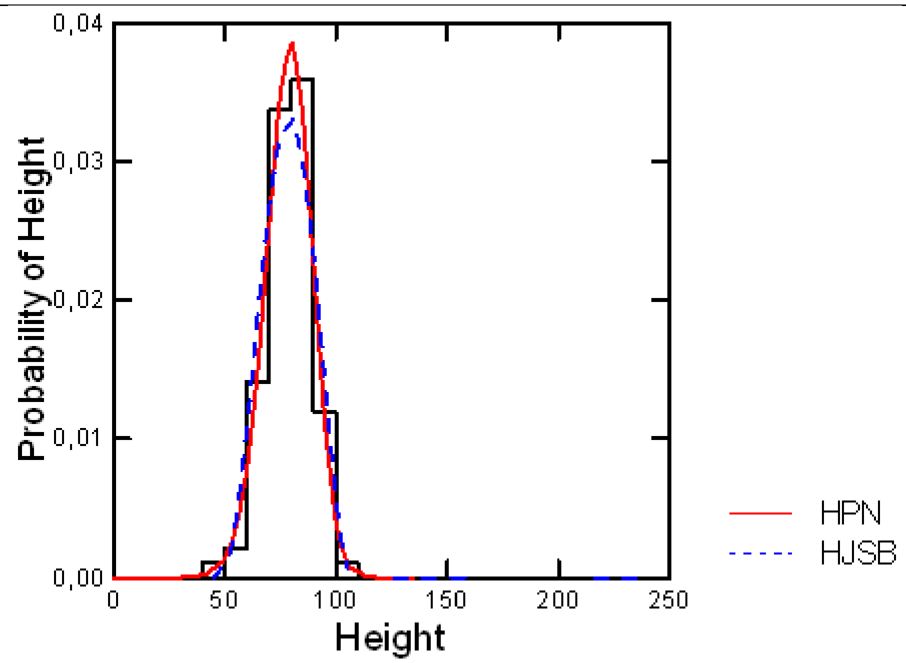


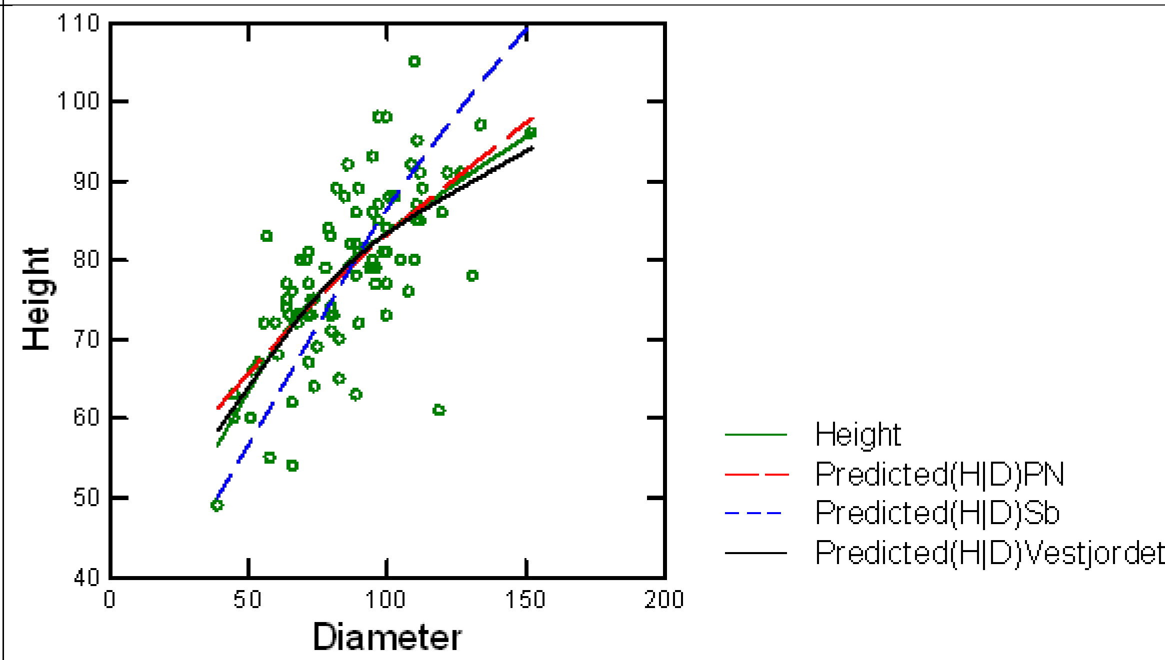




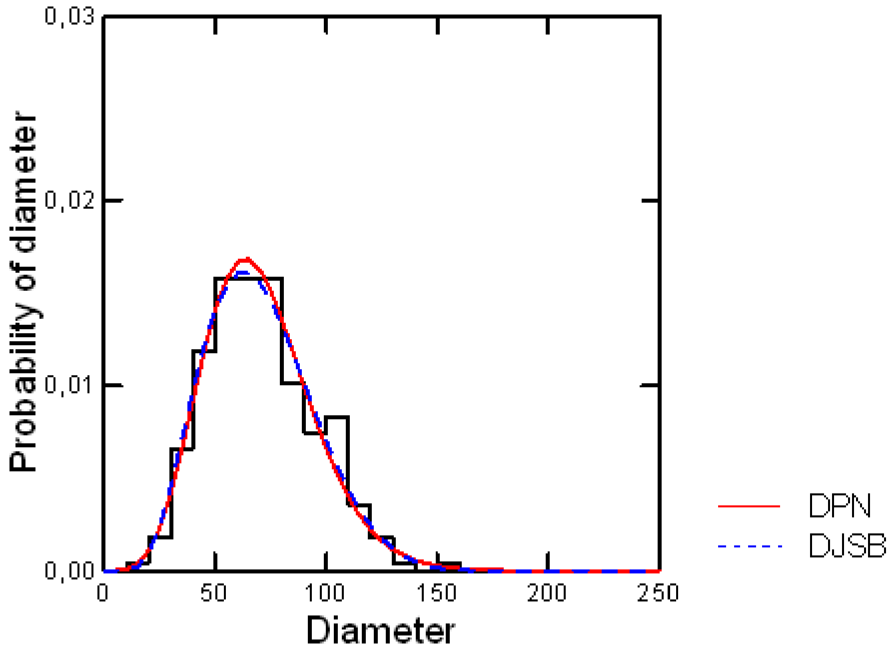
Results (↓) for STAND = 30

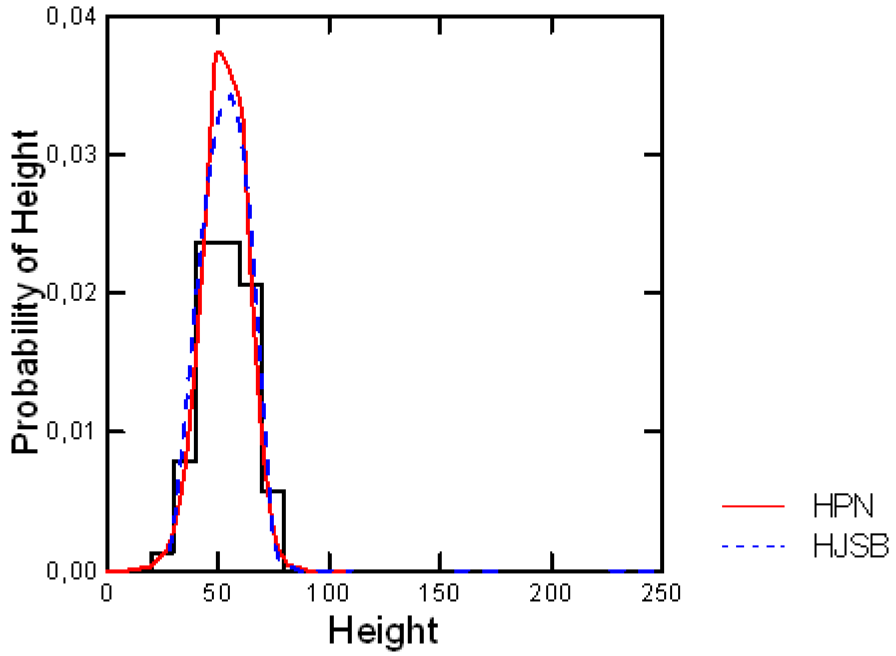


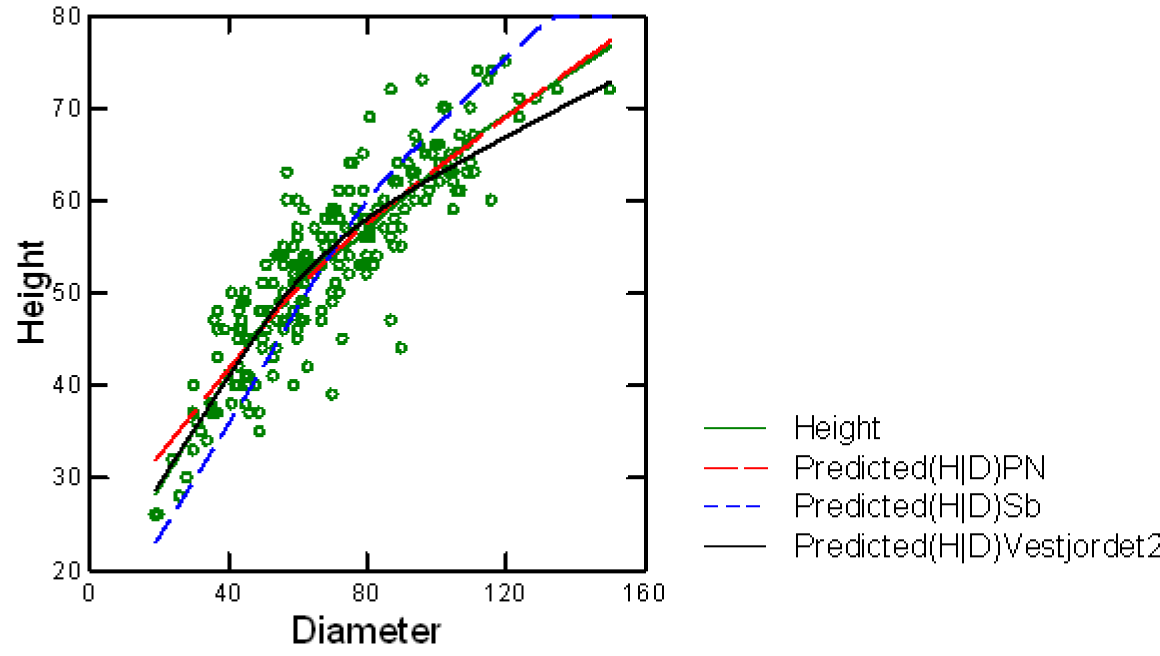




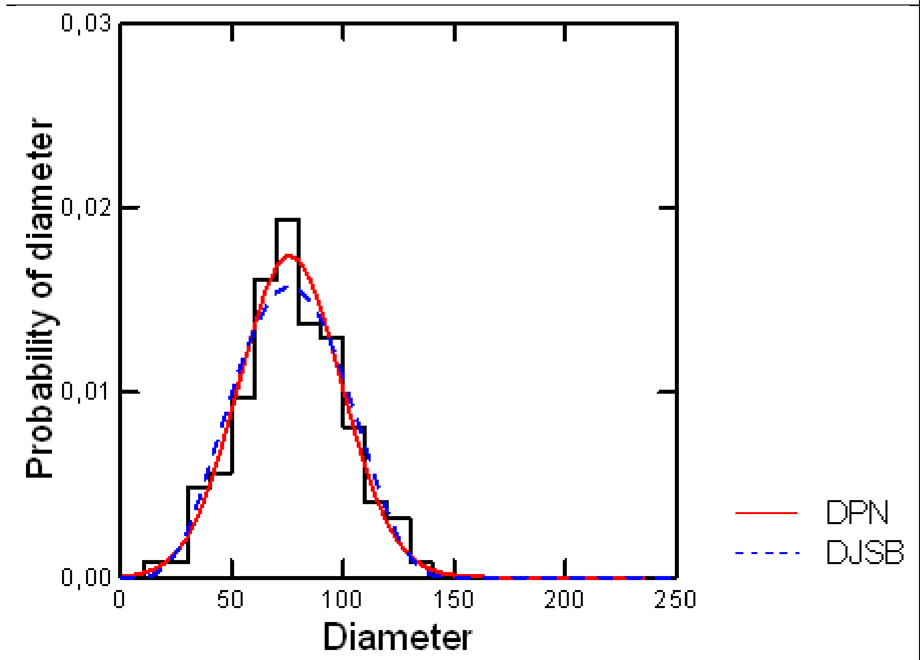
Results (↓) for STAND = 31

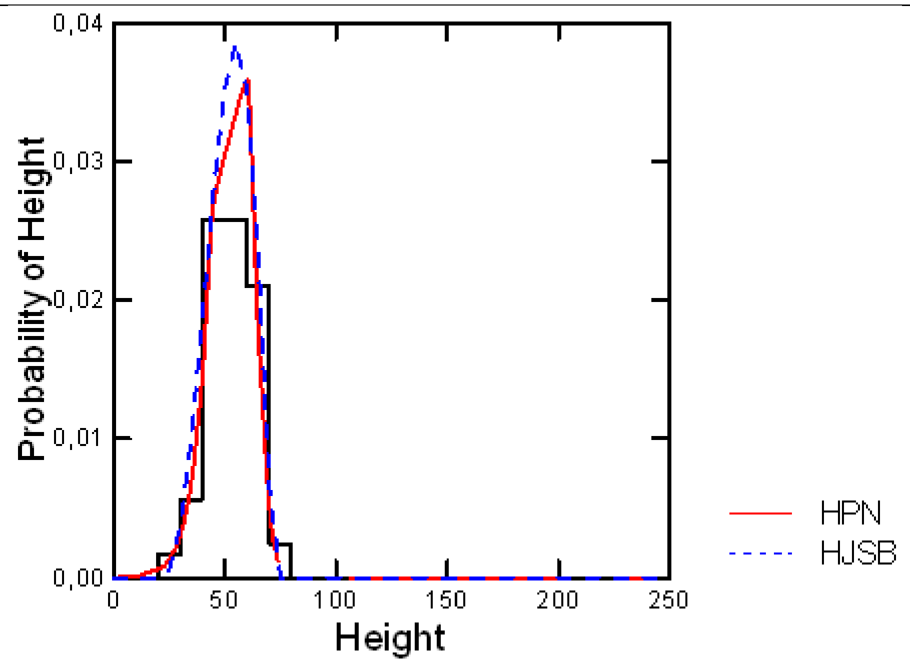


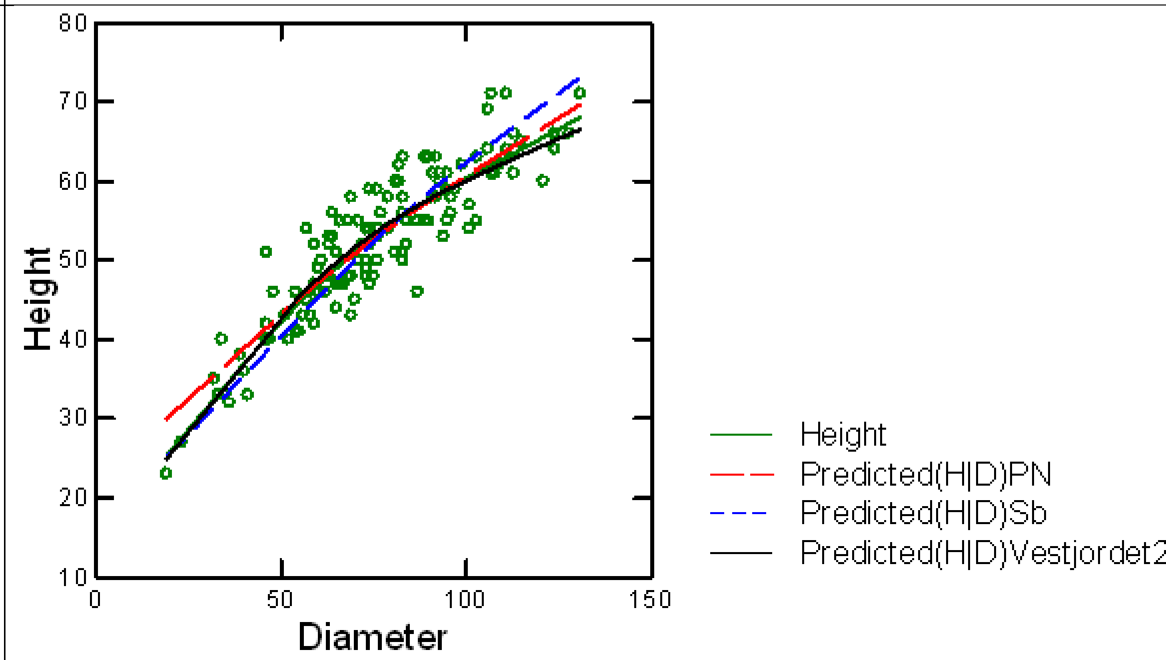




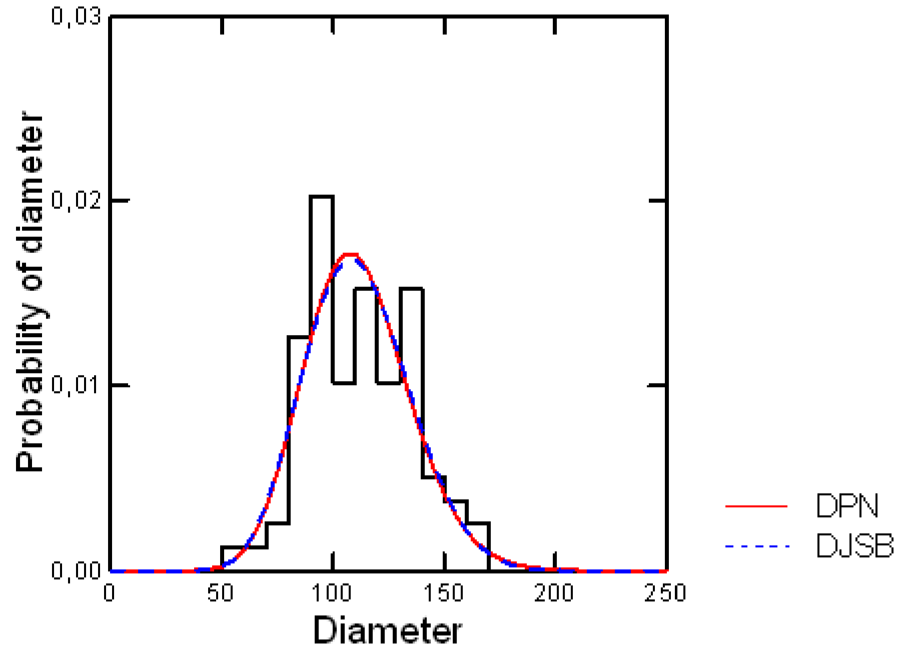
Results (↓) for STAND = 32

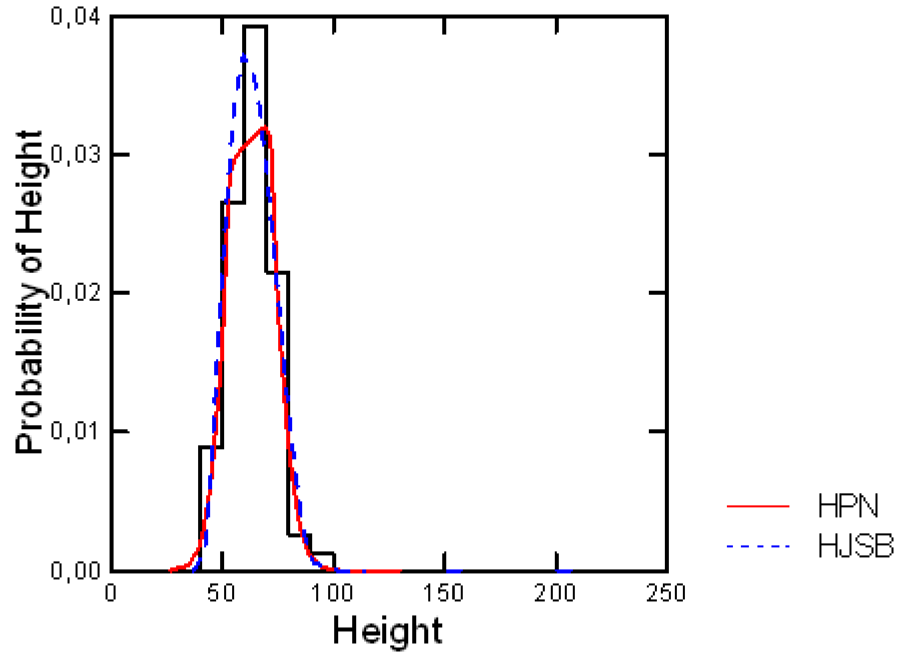


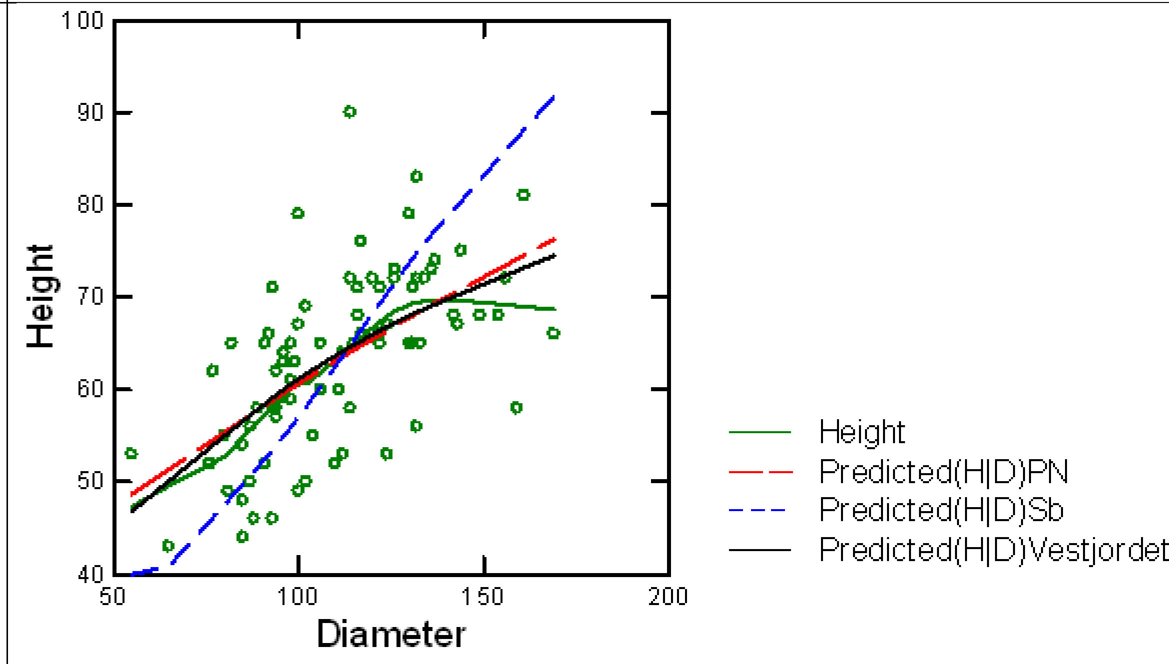




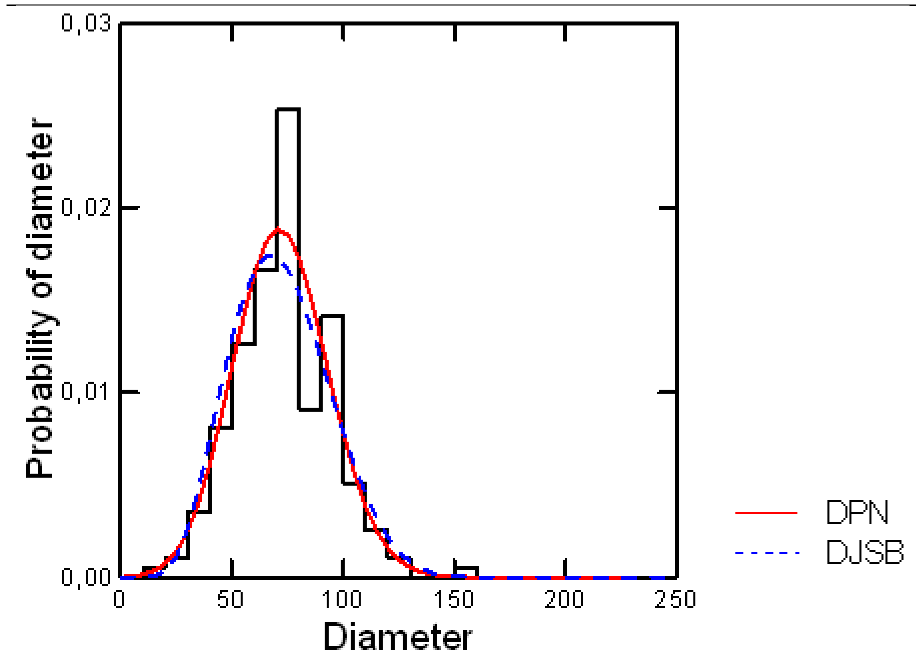
Results (↓) for STAND = 33

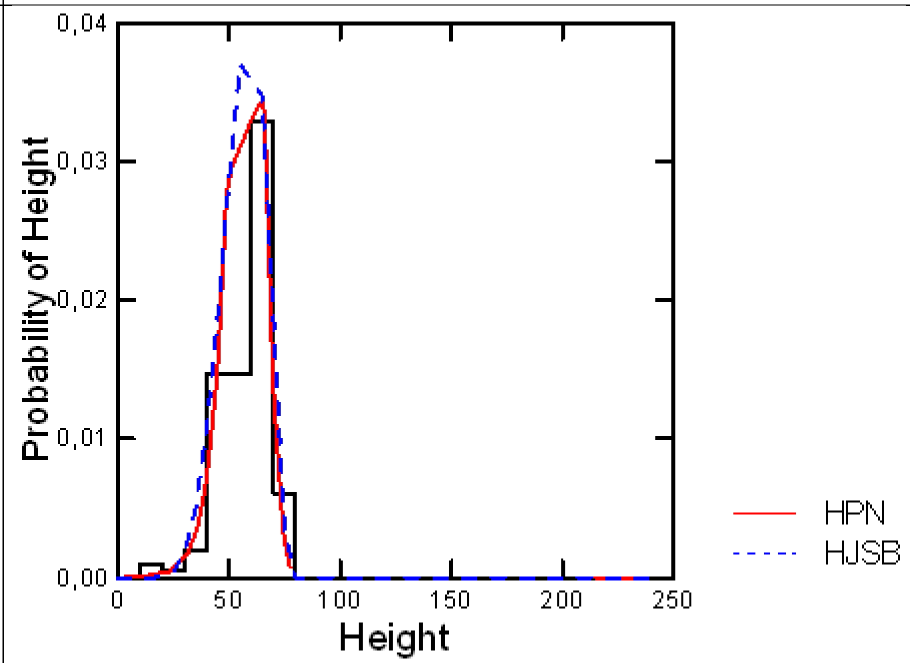


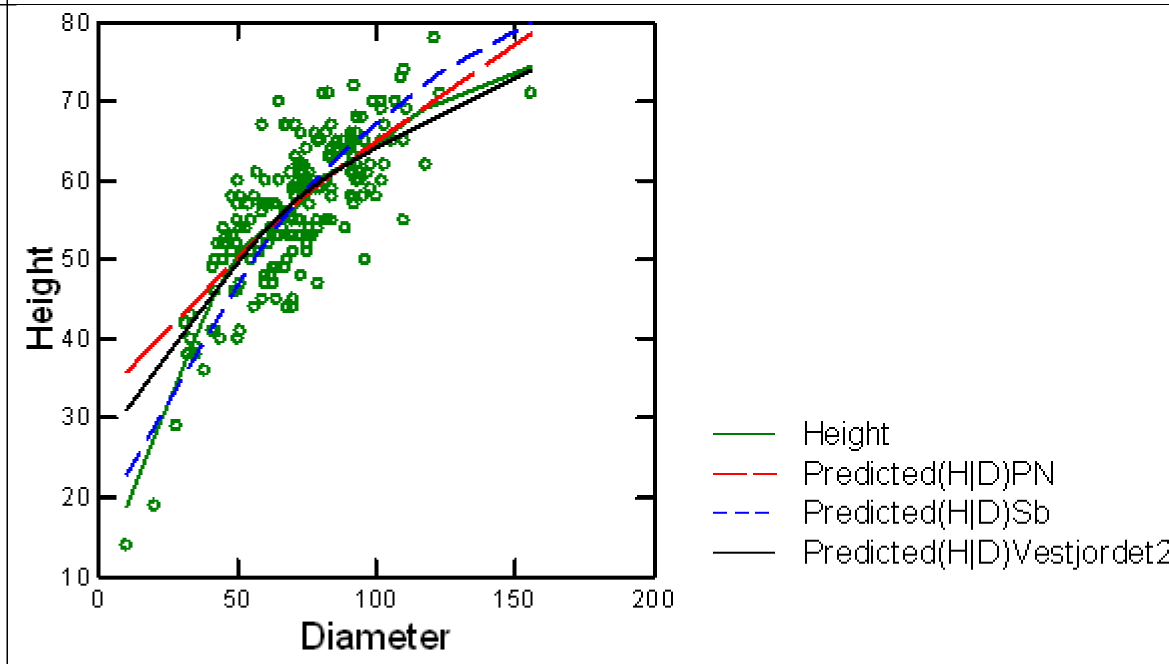




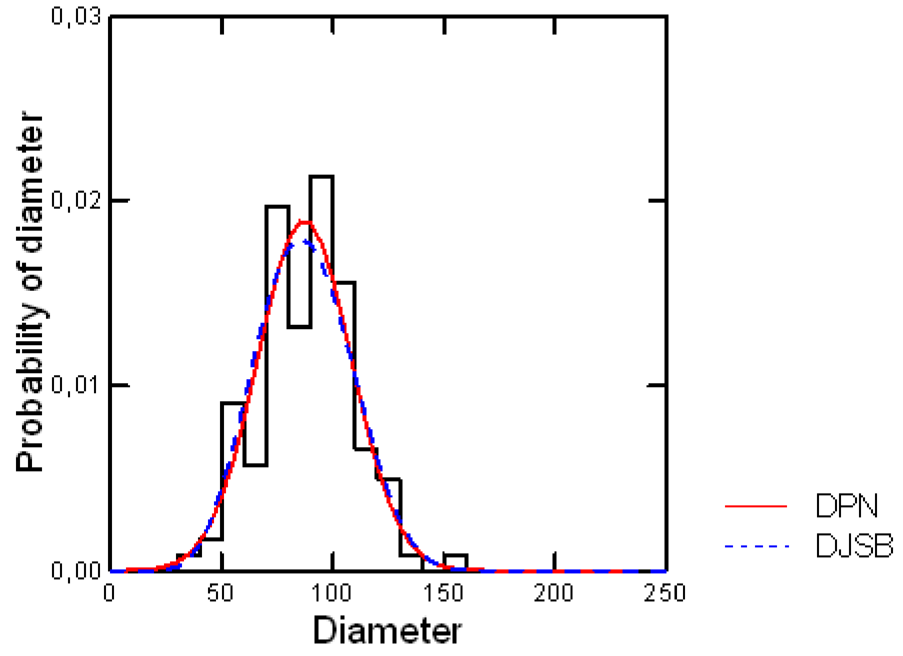
Results (↓) for STAND = 34

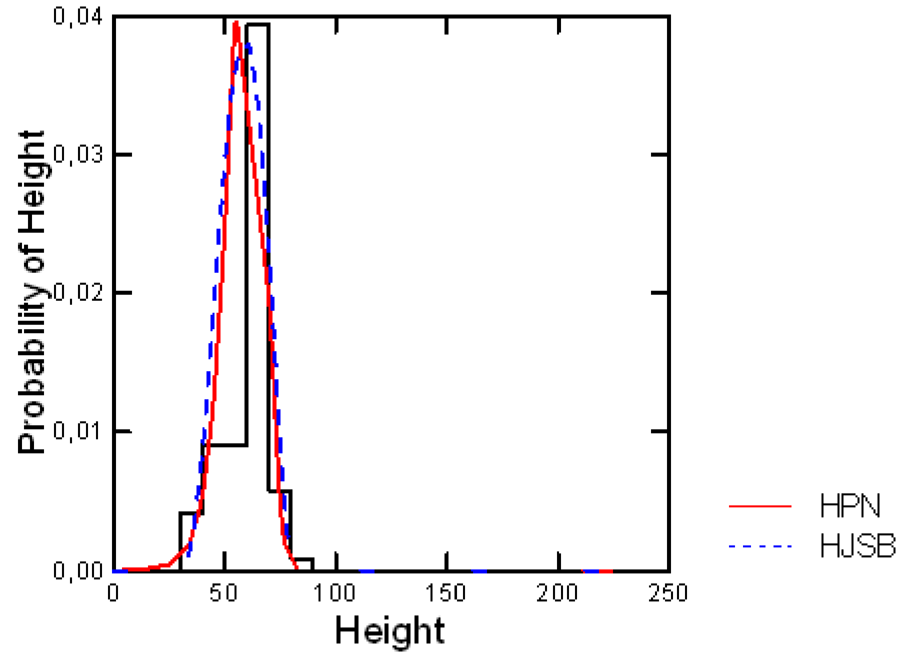


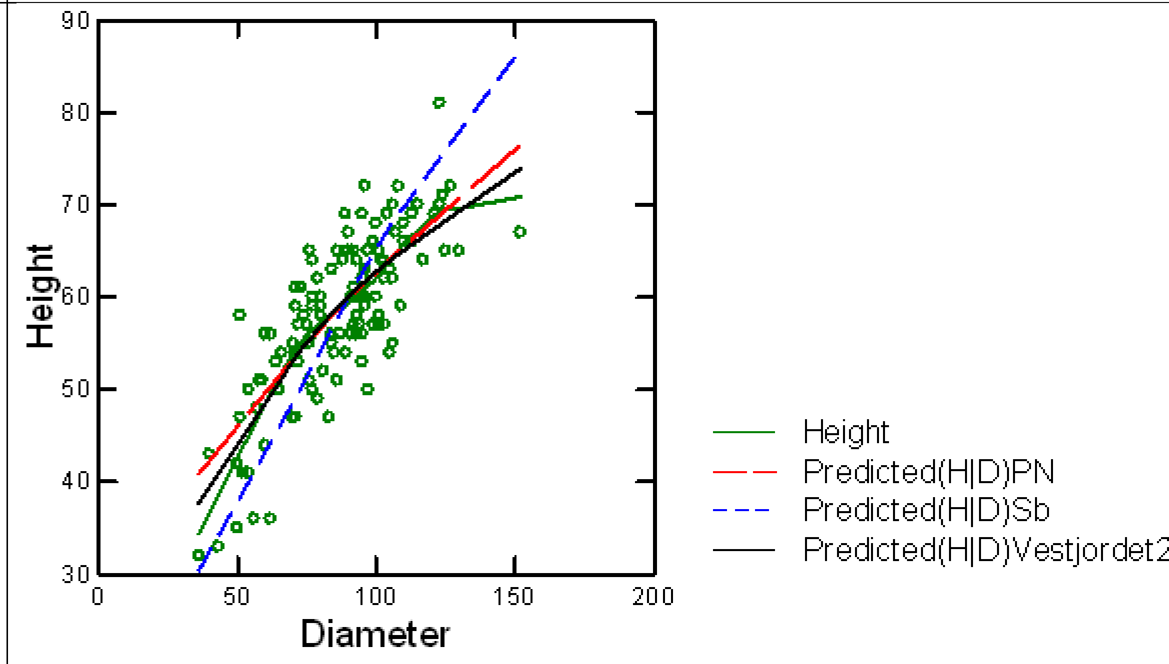




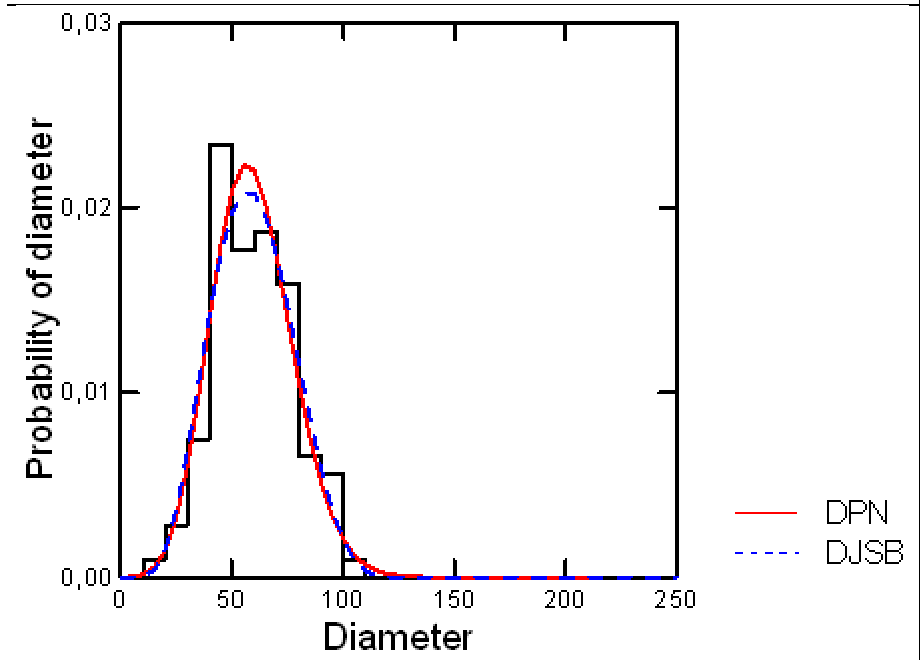
Results (↓) for STAND = 35

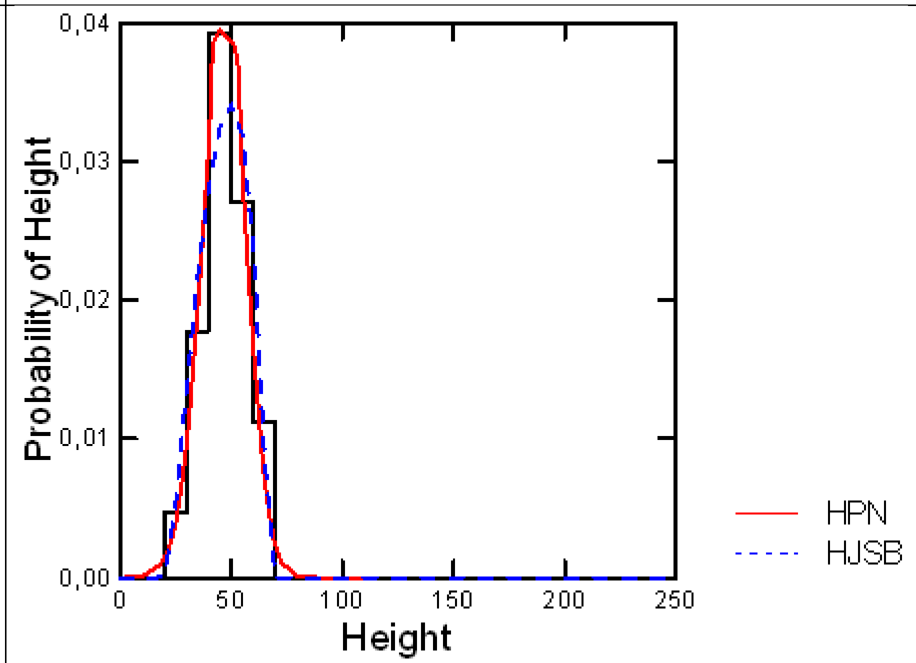


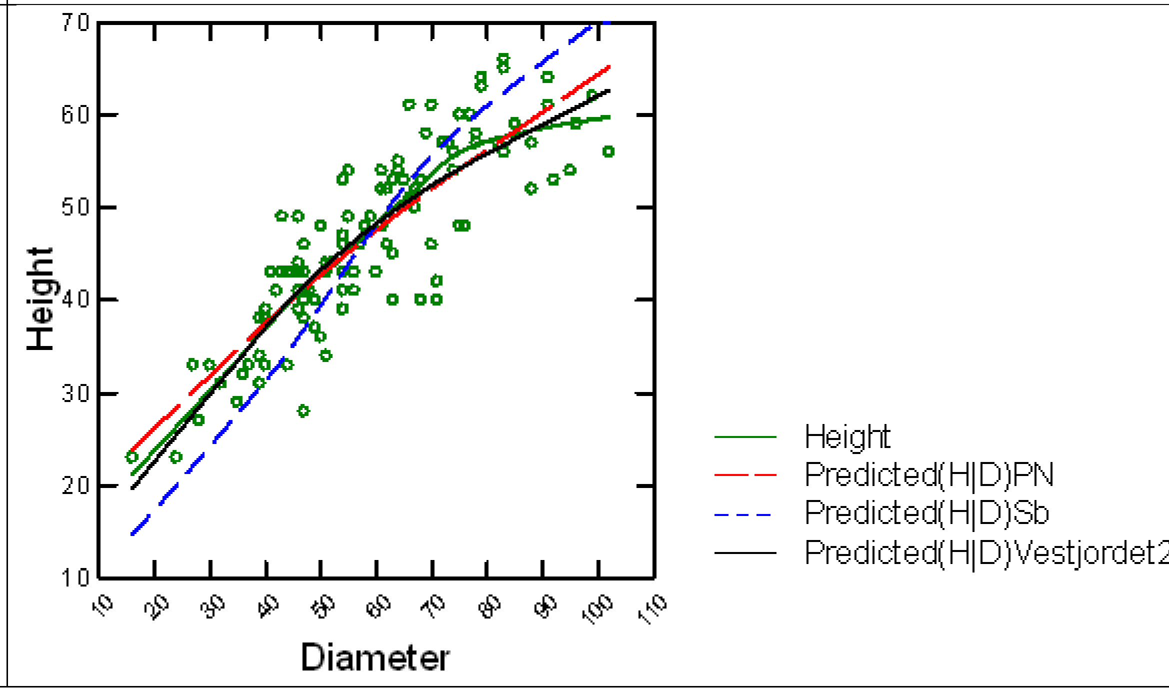




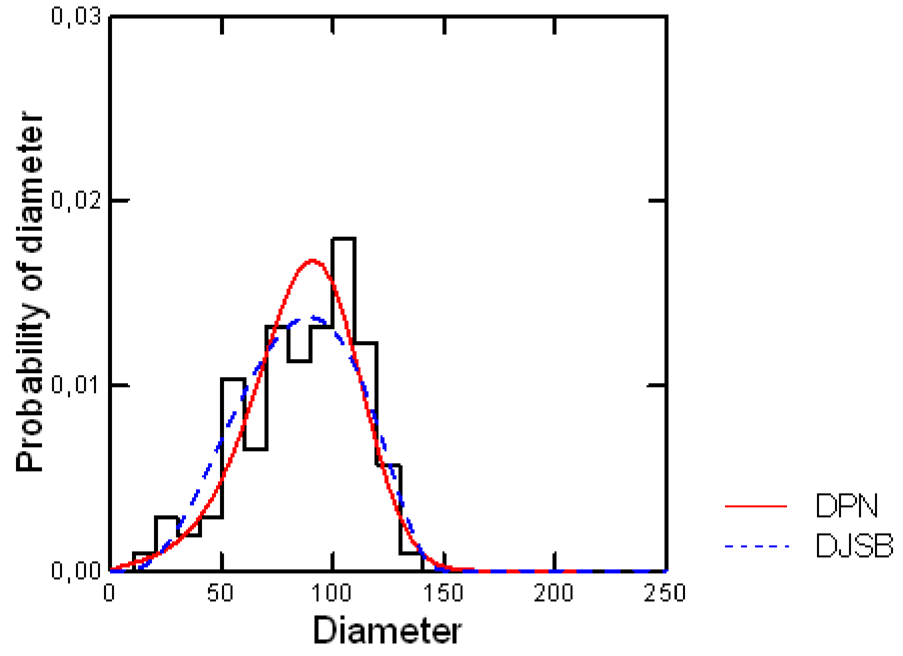
Results (↓) for STAND = 36

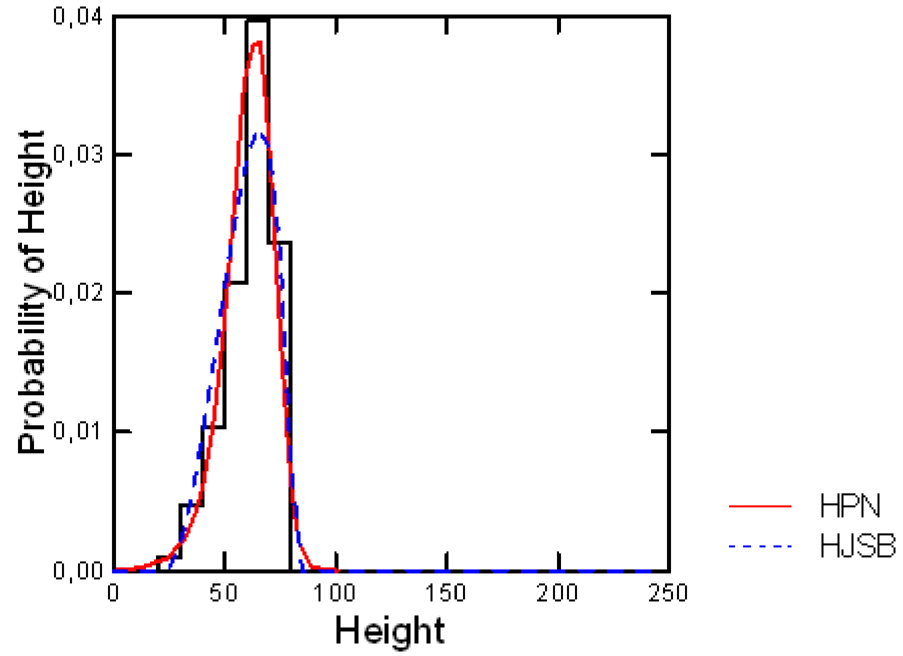


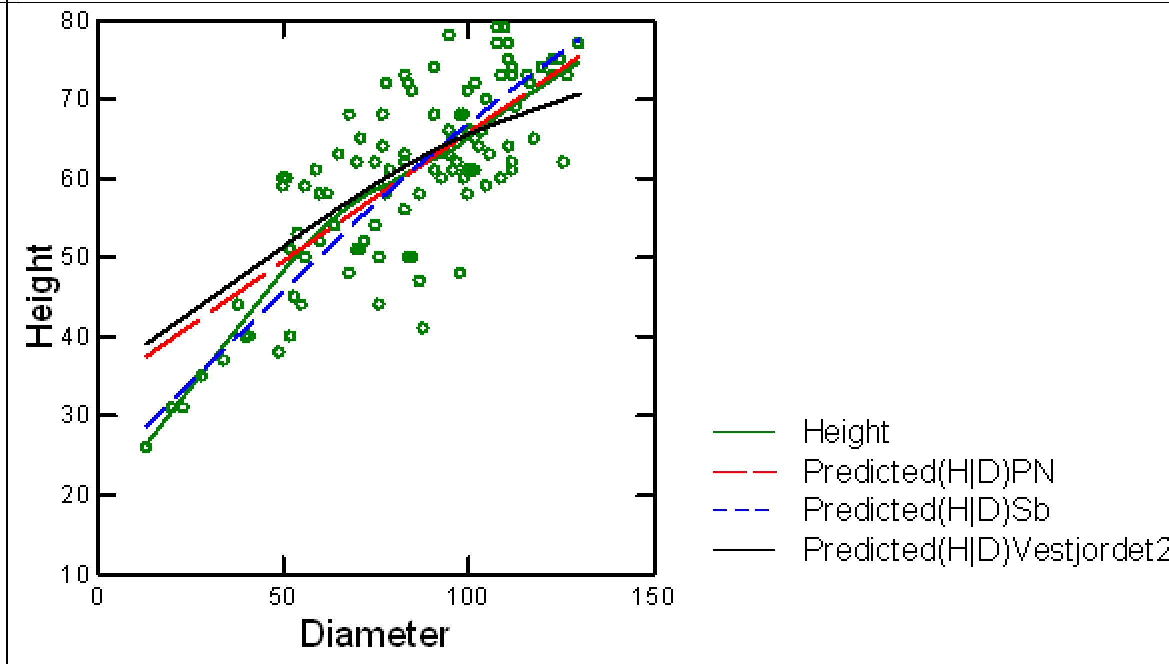




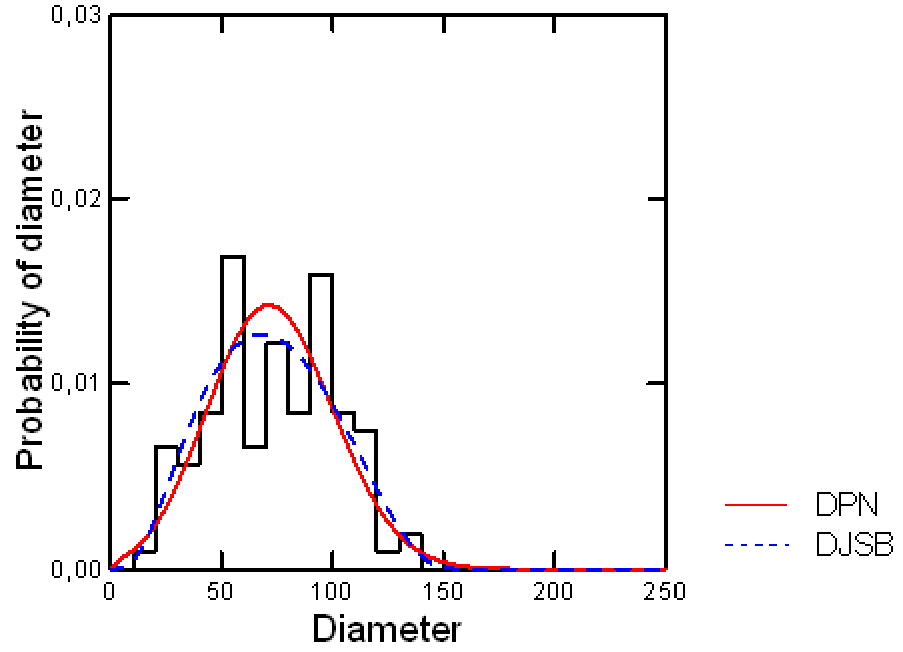
Results (↓) for STAND = 37

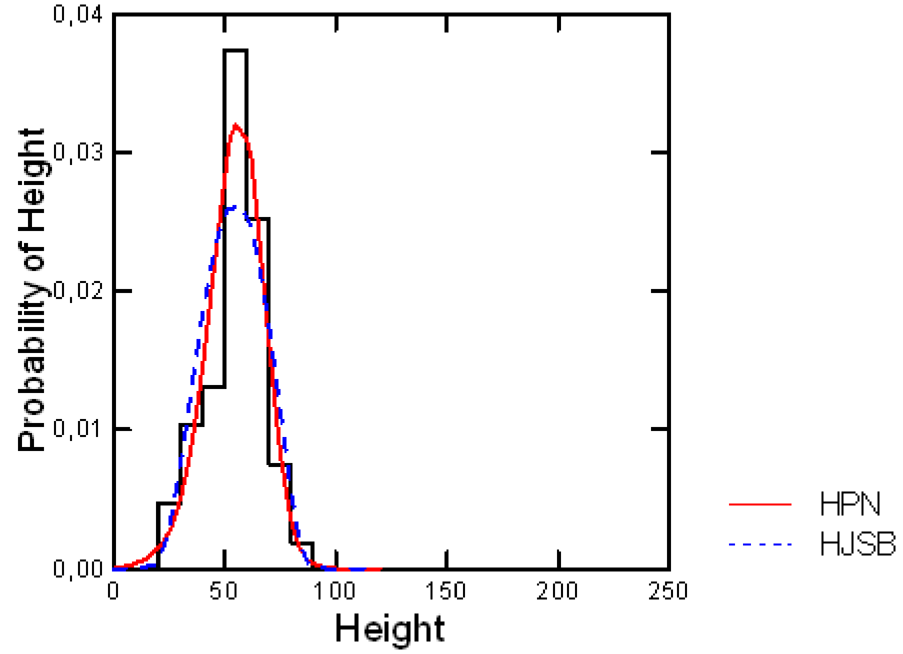


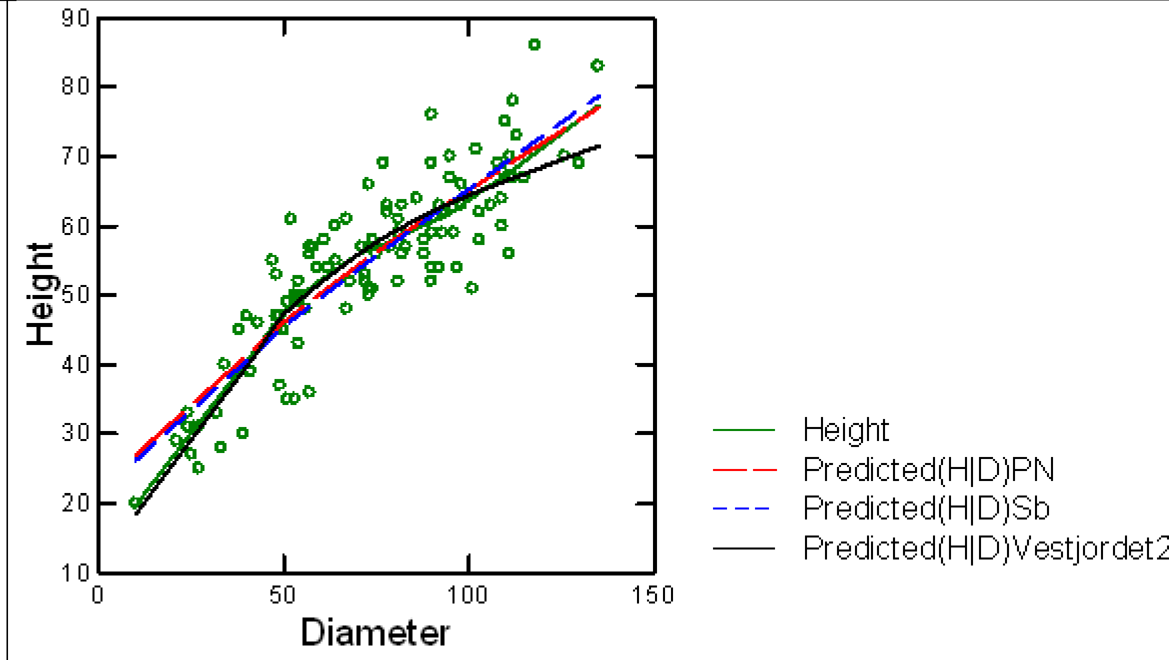




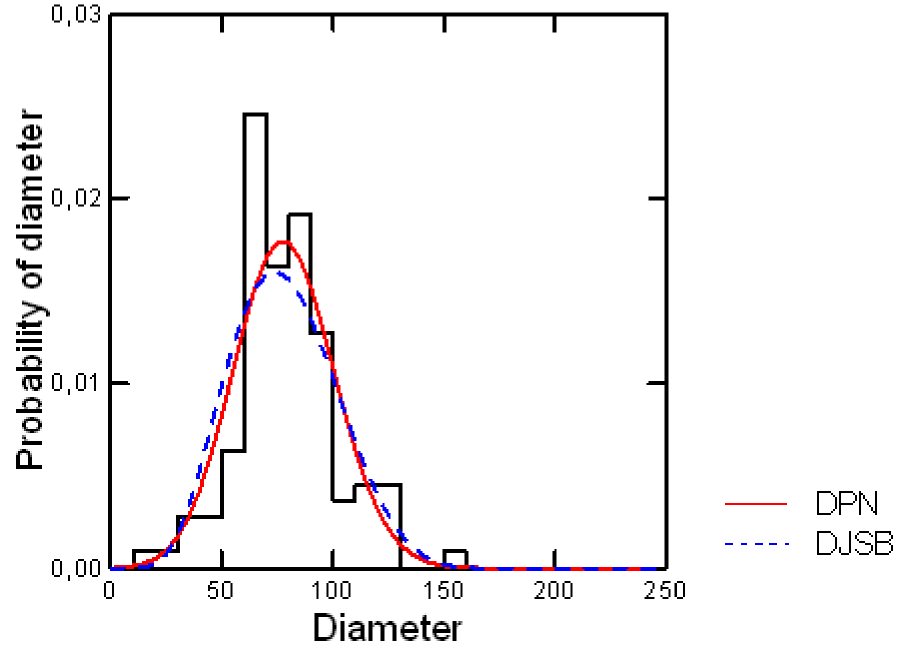
Results (↓) for STAND = 38

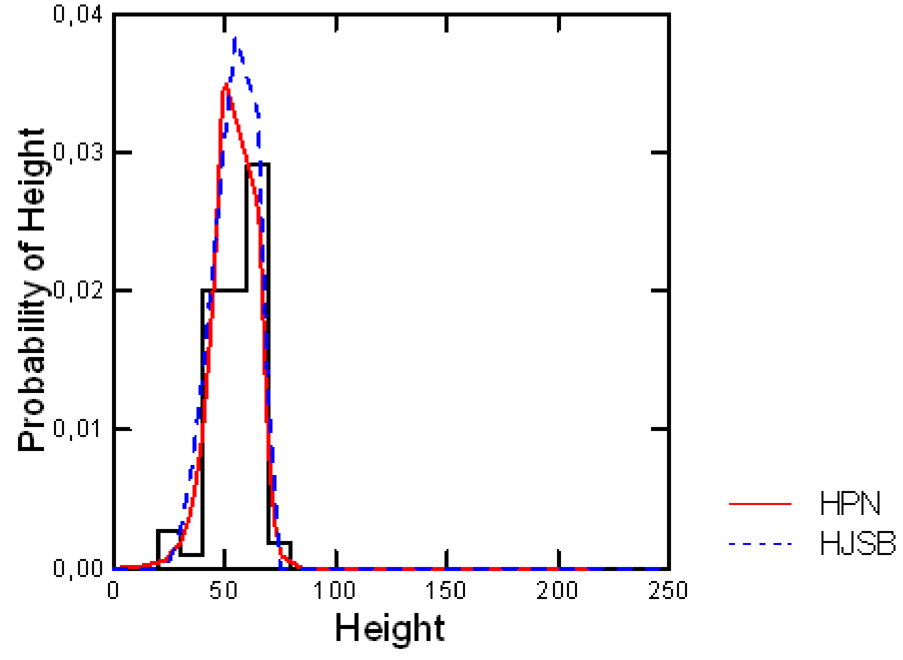


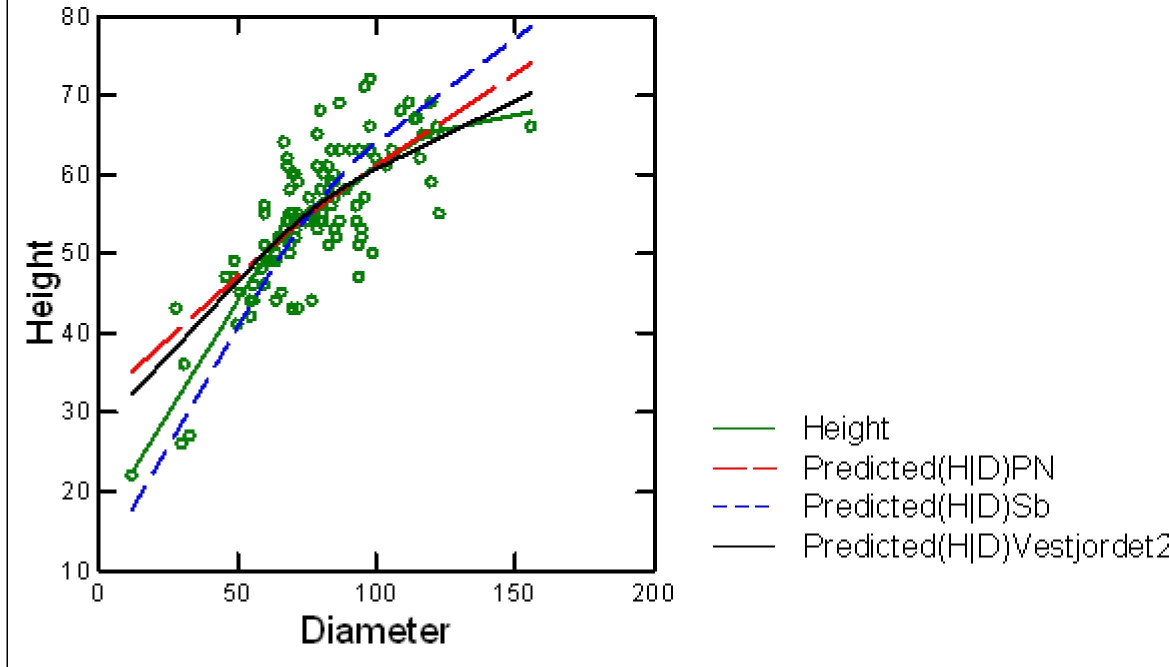




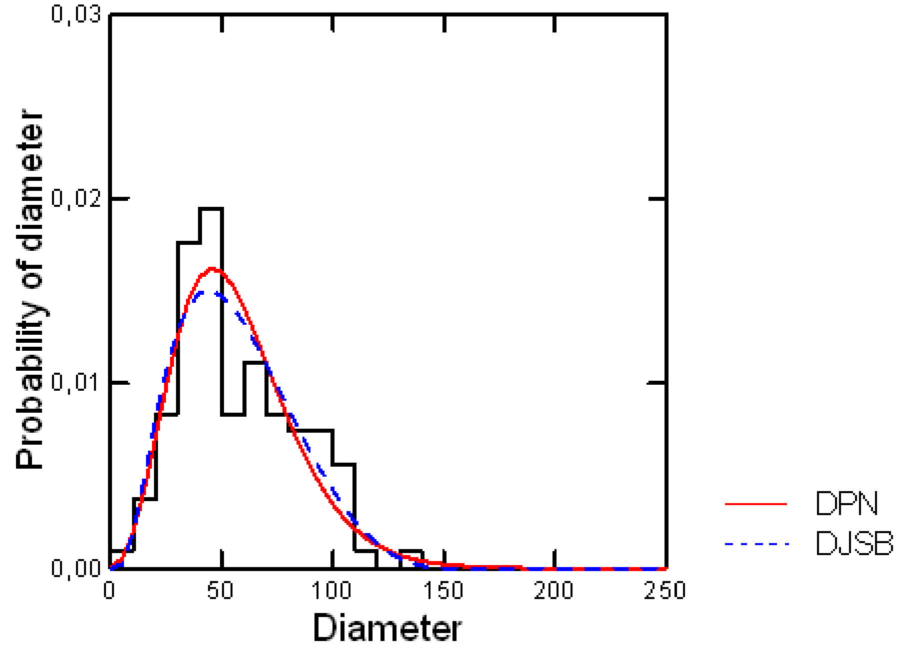
Results (↓) for STAND = 39

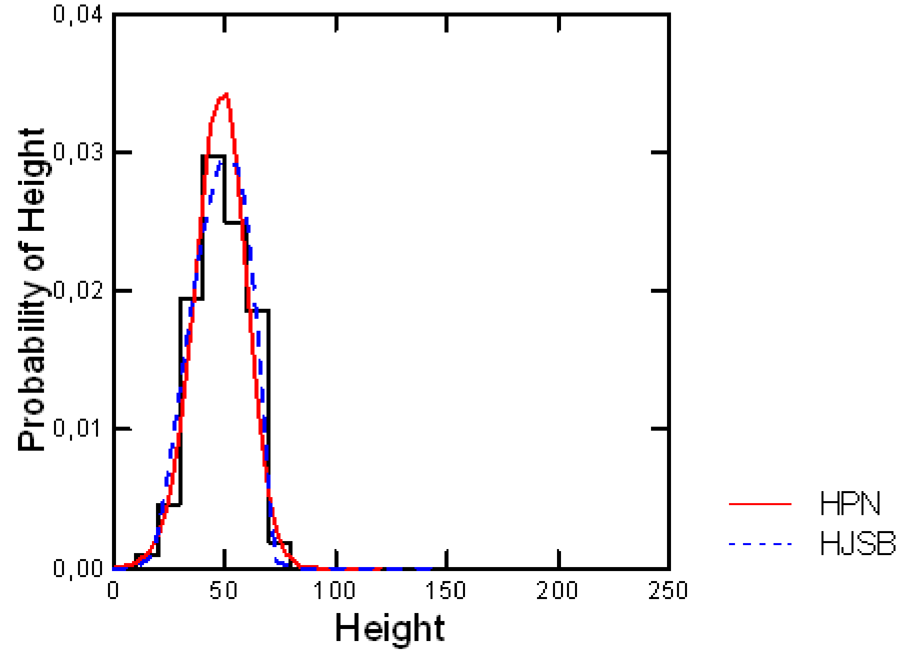


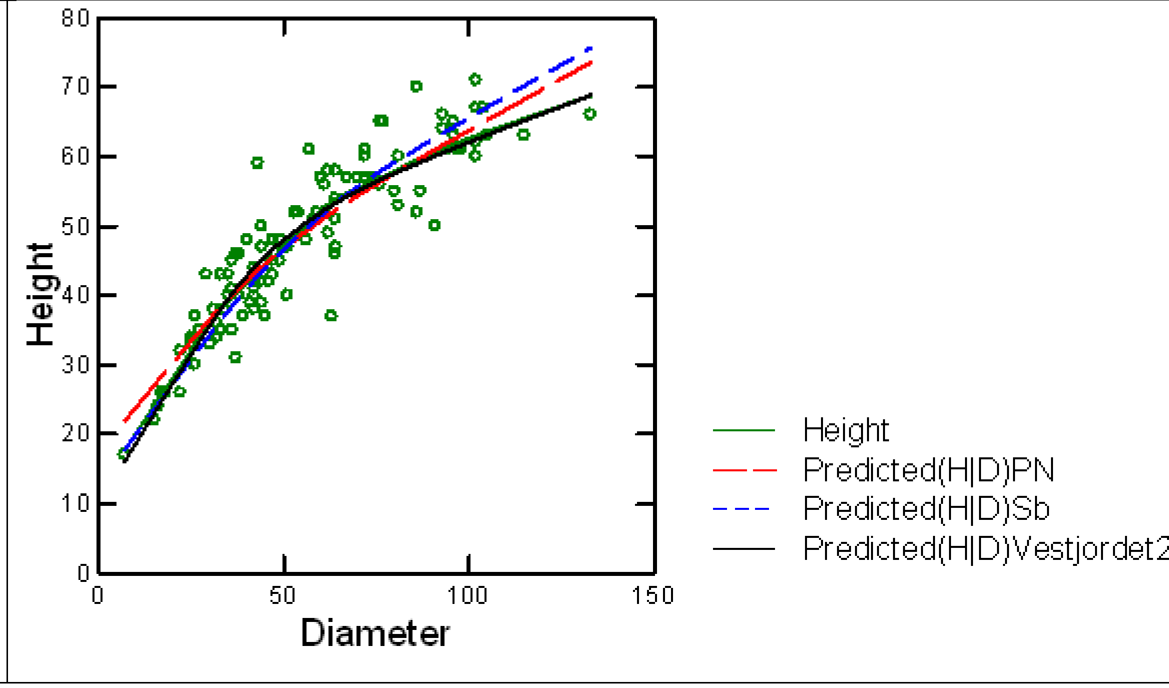




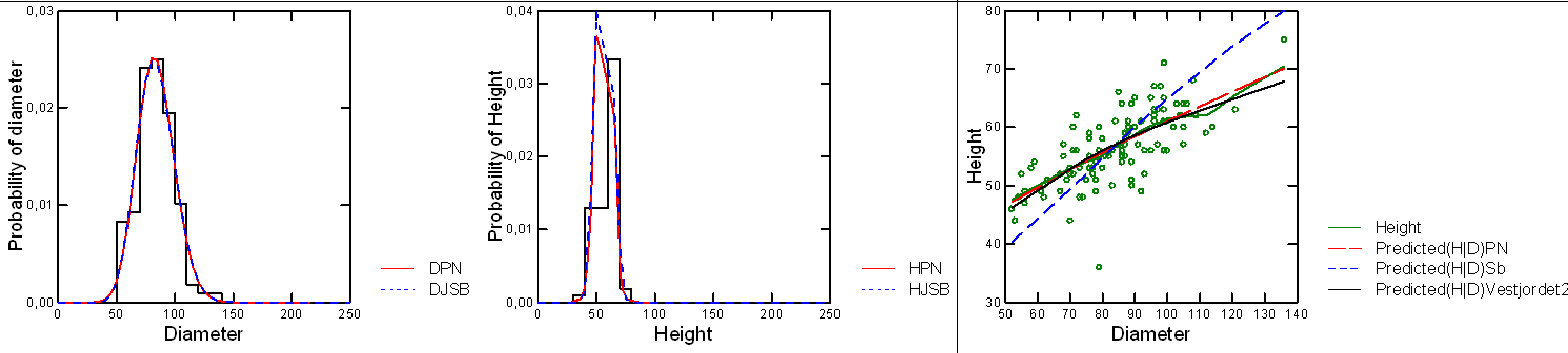
Results (↓) for STAND = 40



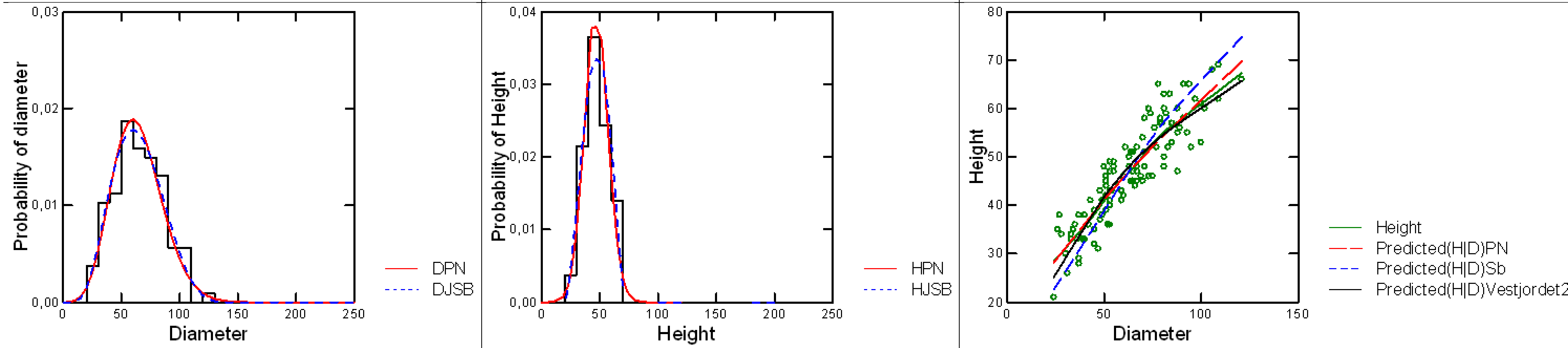




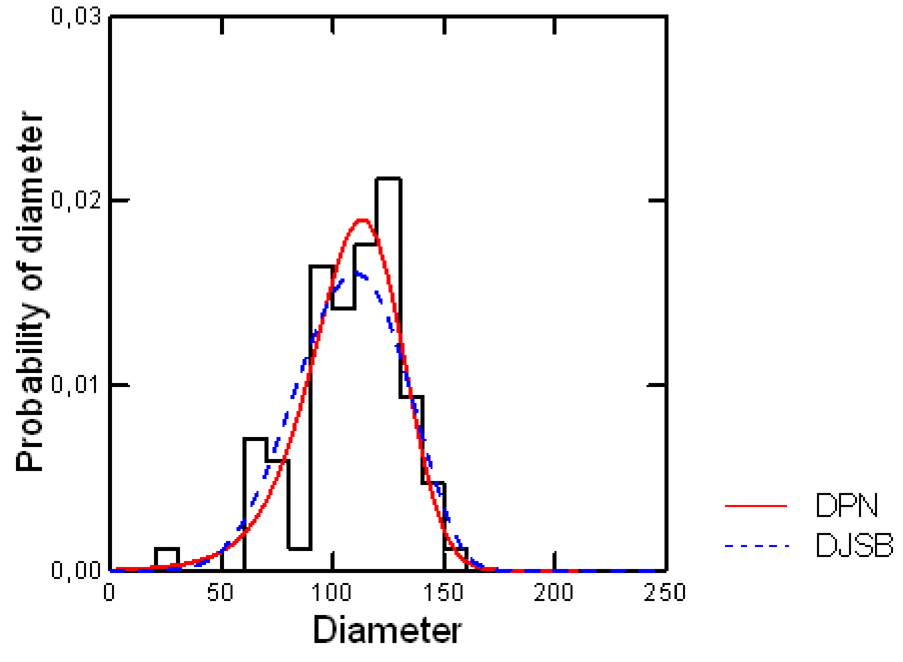
Results (↓) for STAND = 41

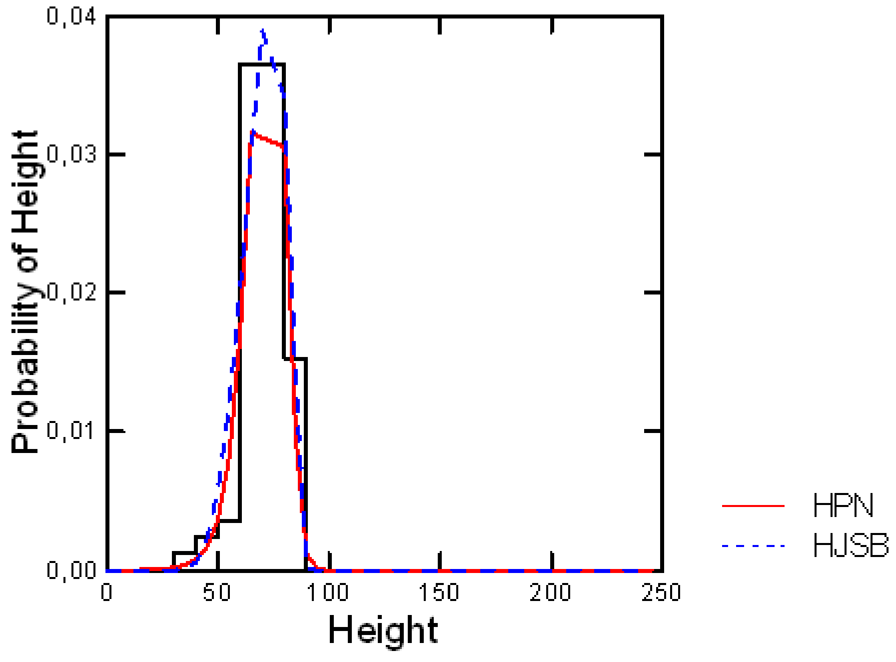


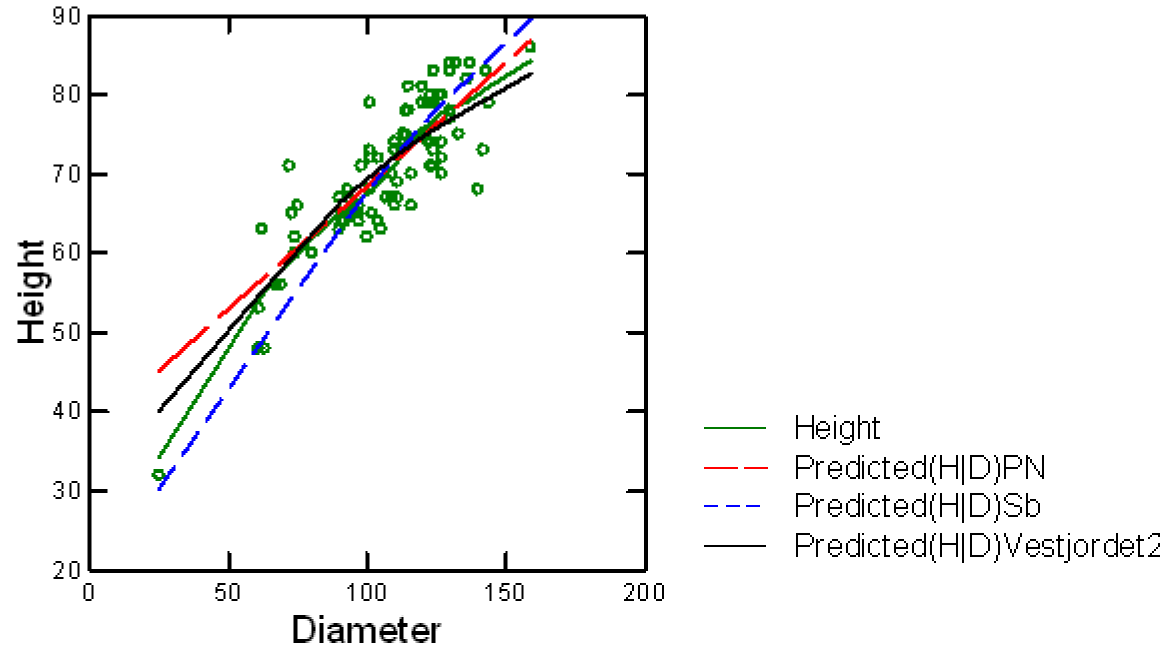
Results (↓) for STAND = 42



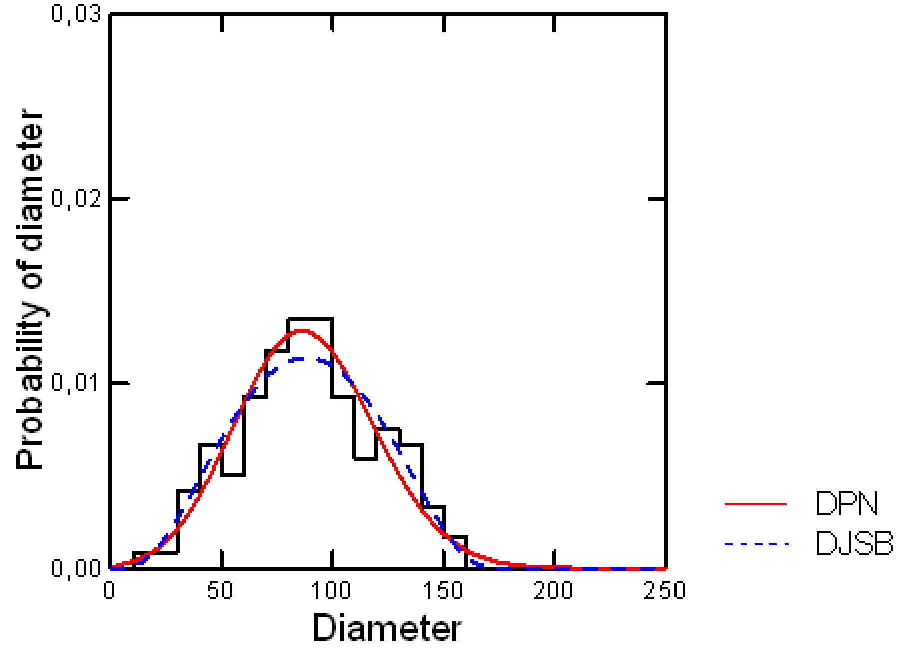
Results (↓) for STAND = 43

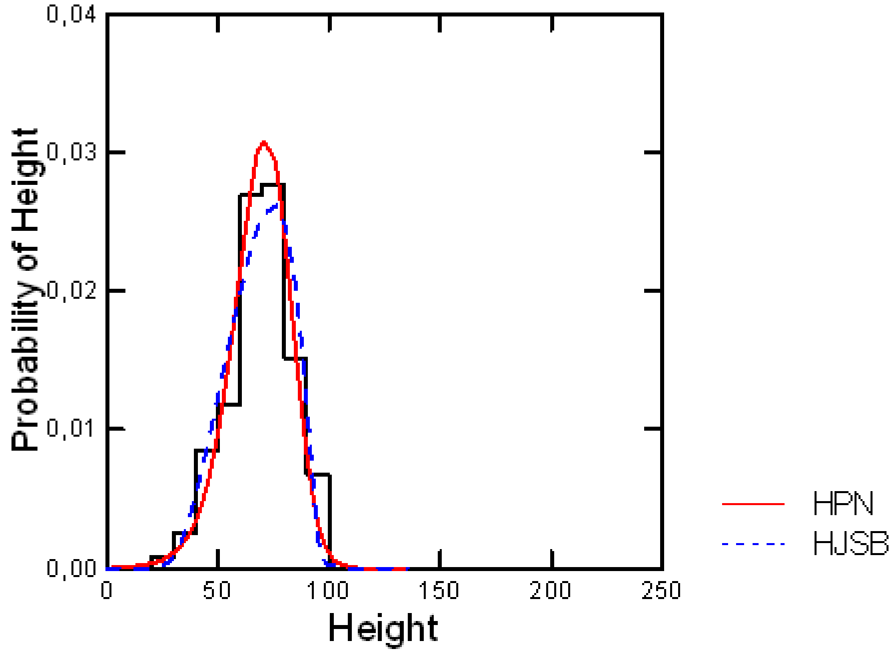


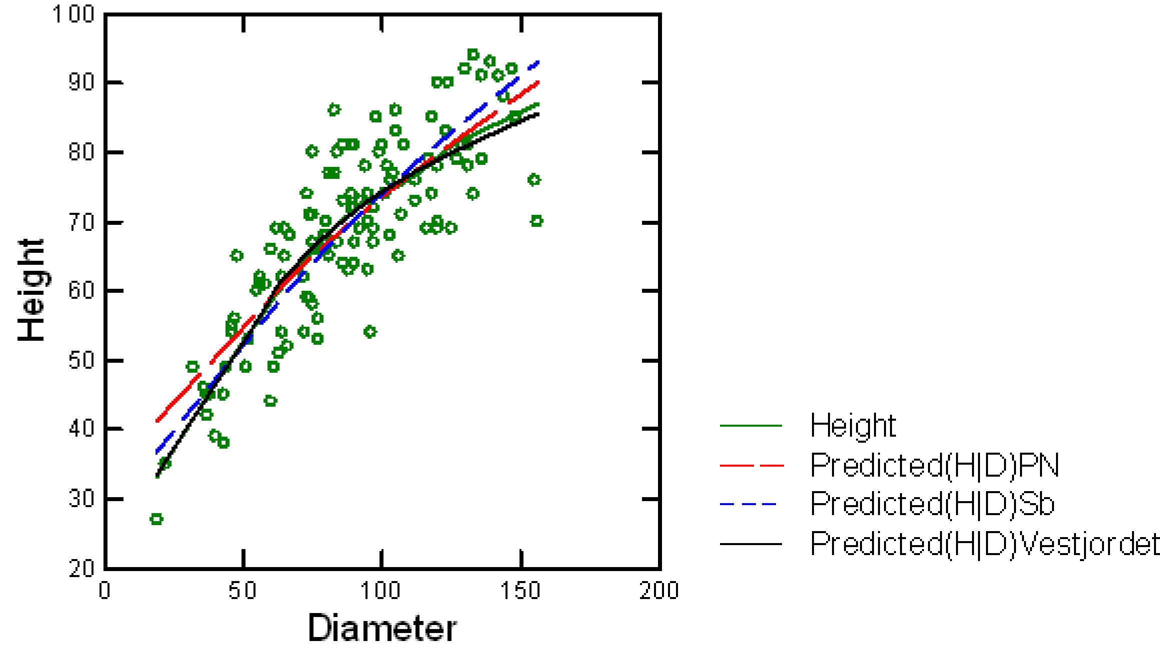


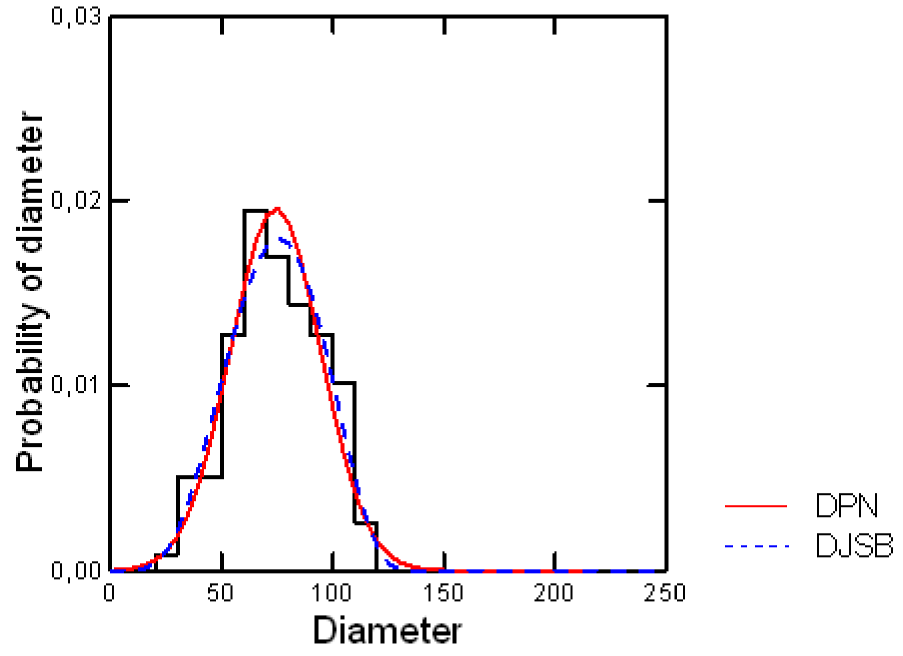


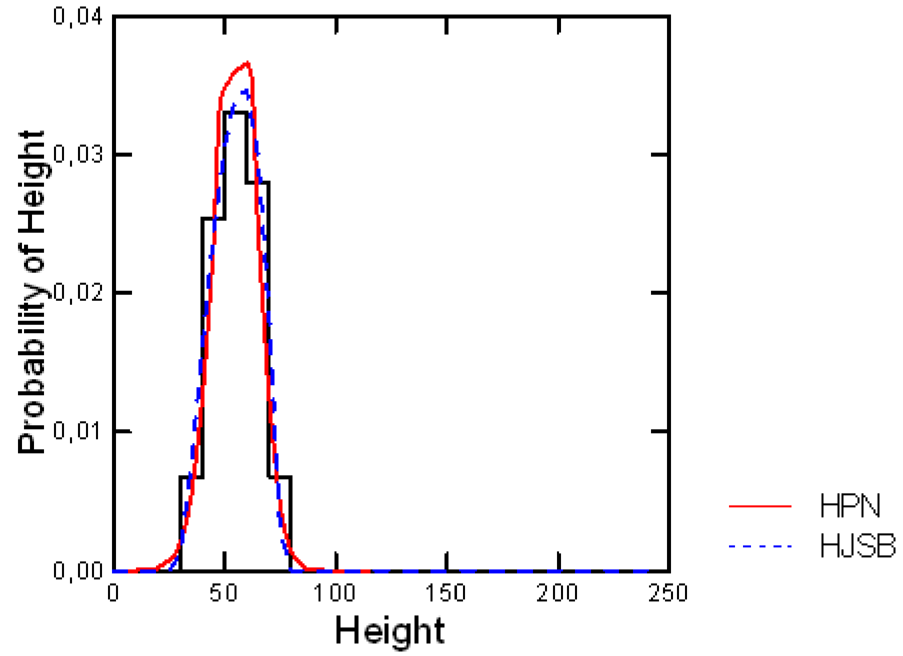
Results (↓) for STAND = 44



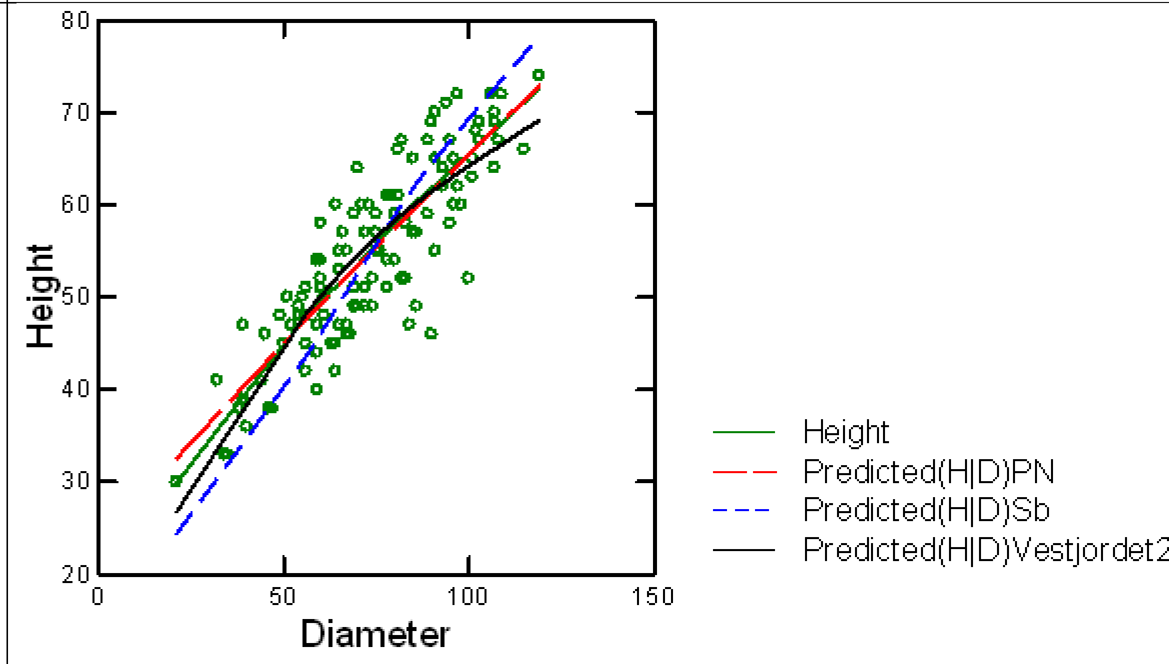


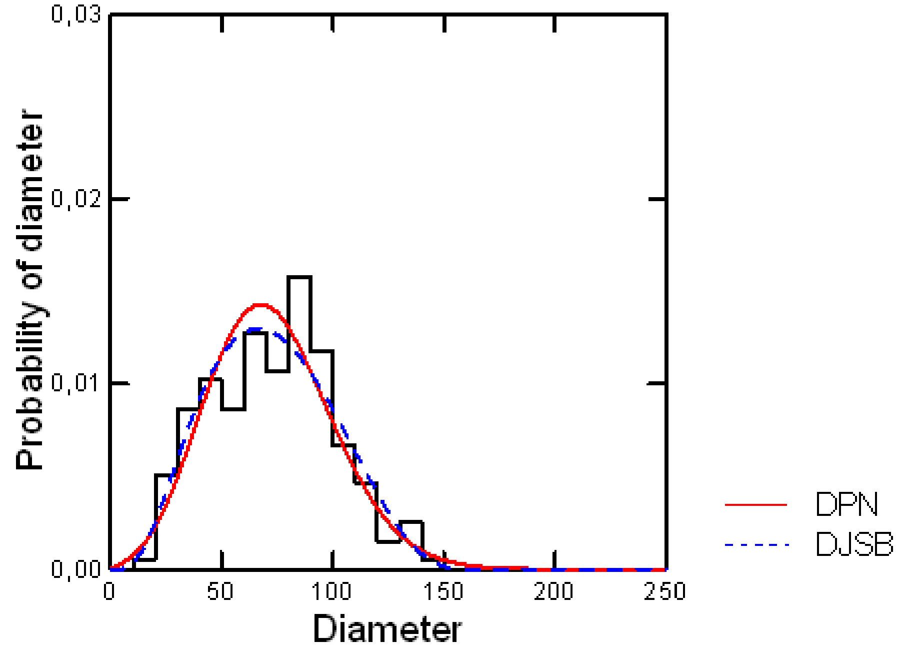


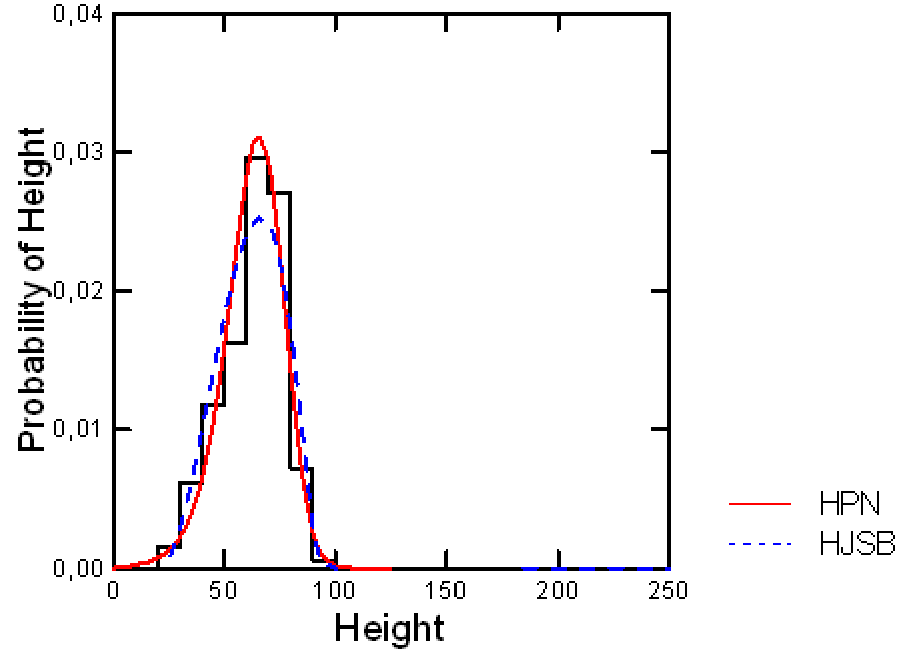




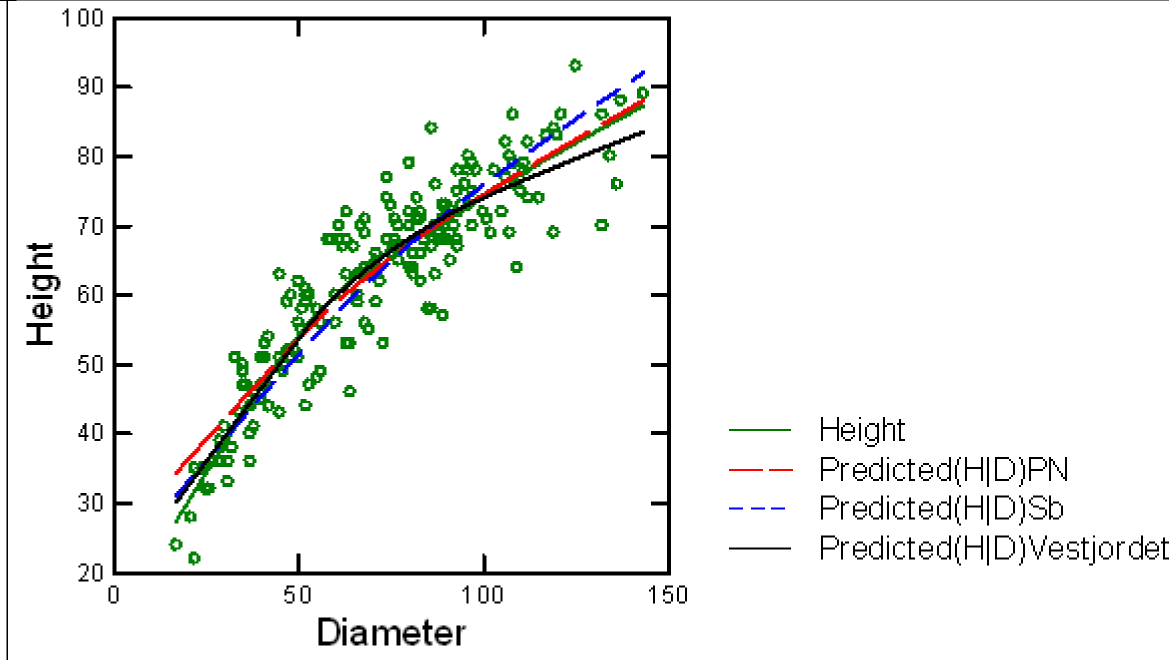
Results (↓) for STAND = 45



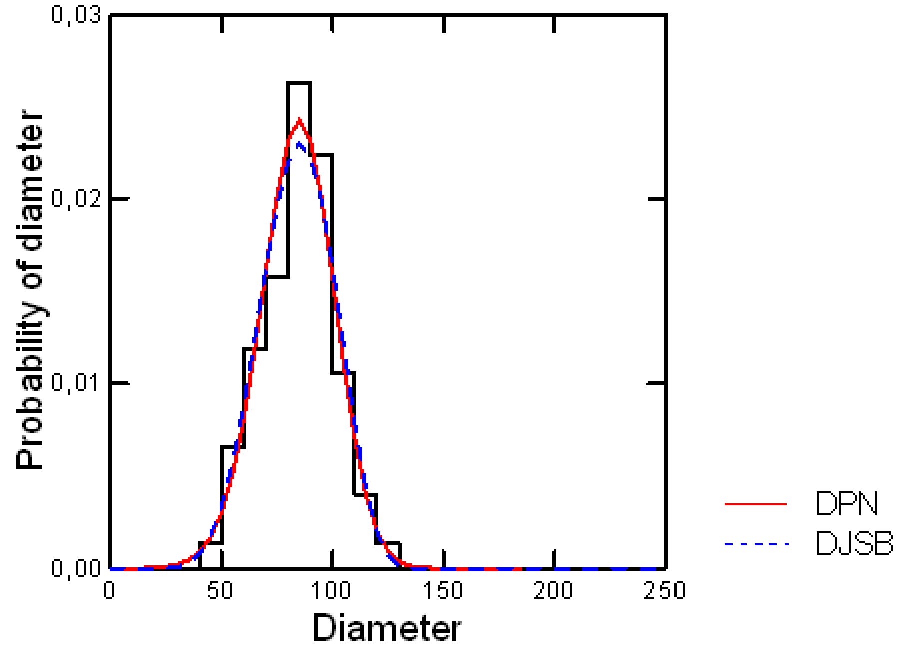


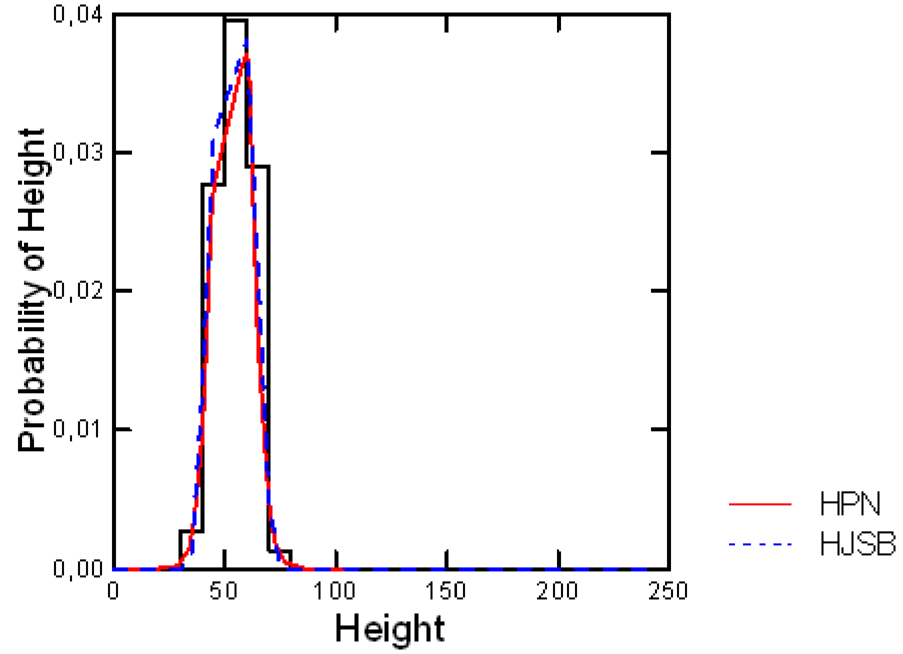


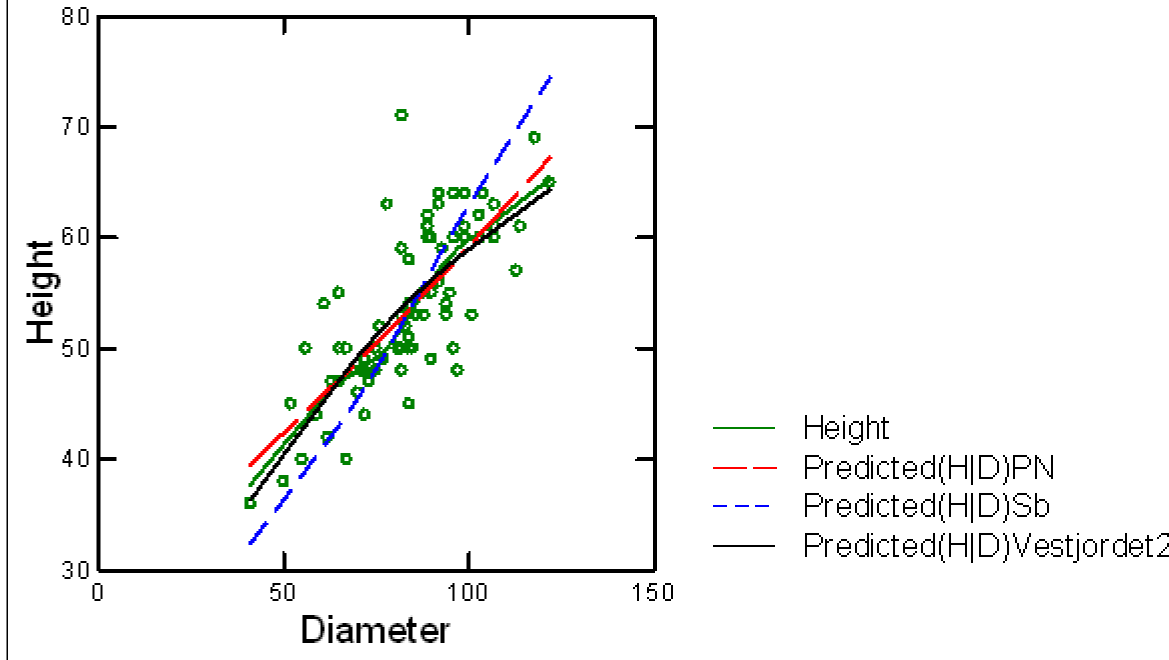
Results (↓) for STAND = 46



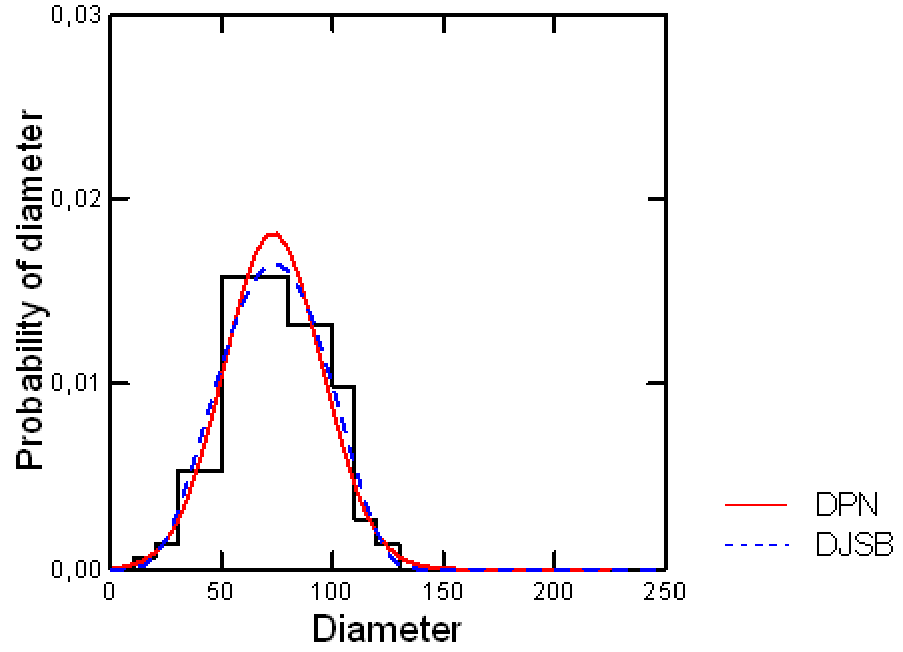
Results (↓) for STAND = 47

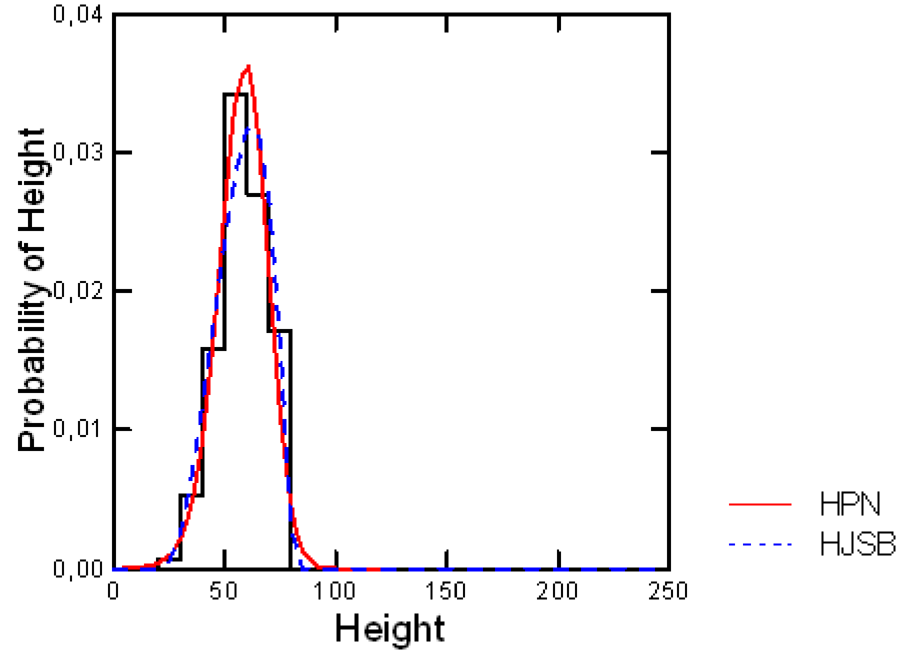


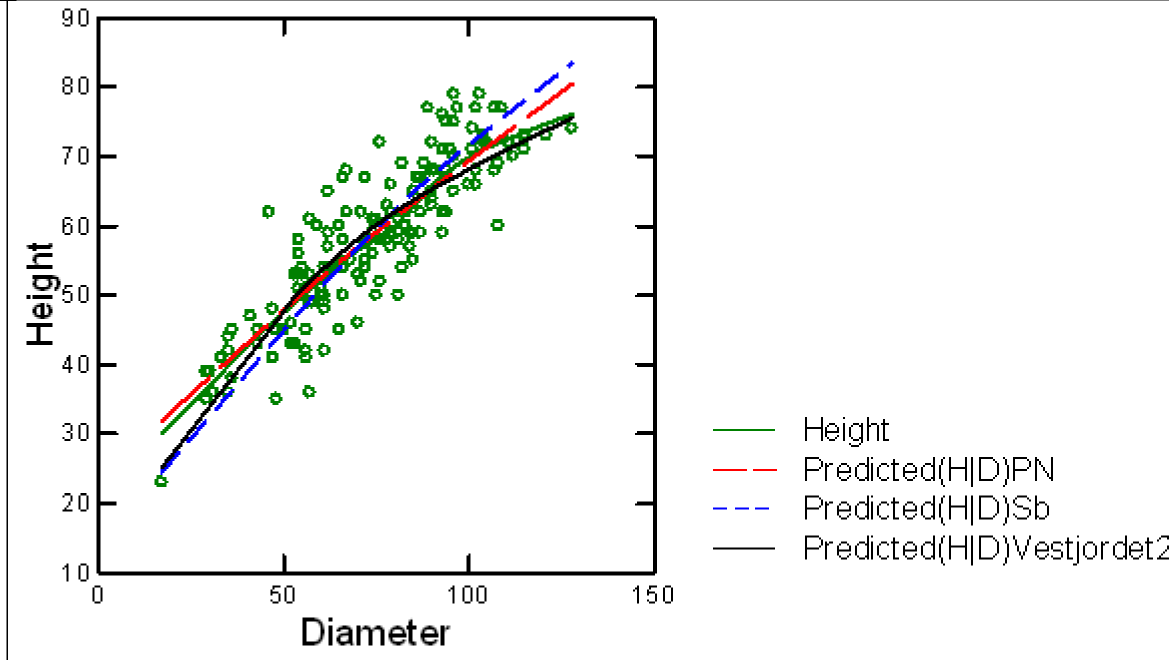




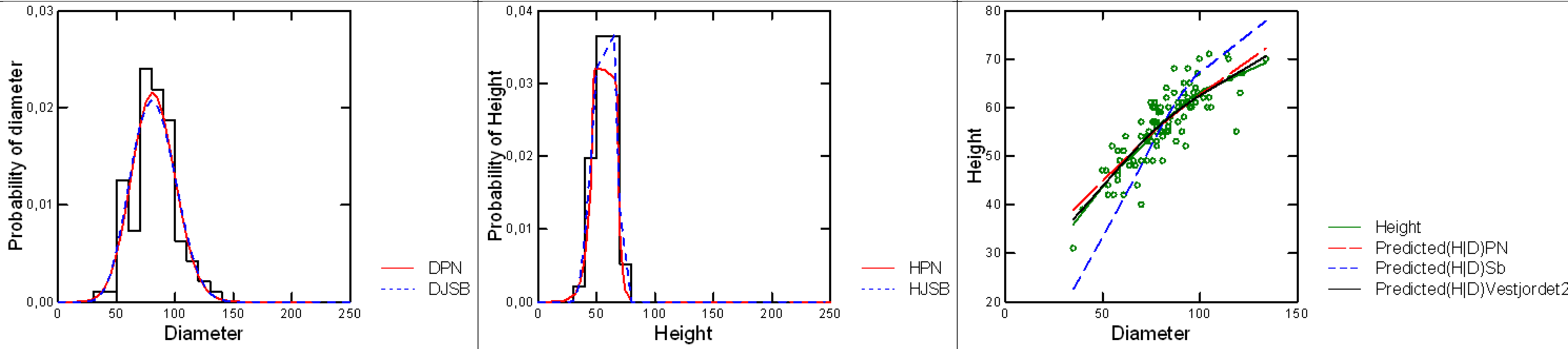
Results (↓) for STAND = 48



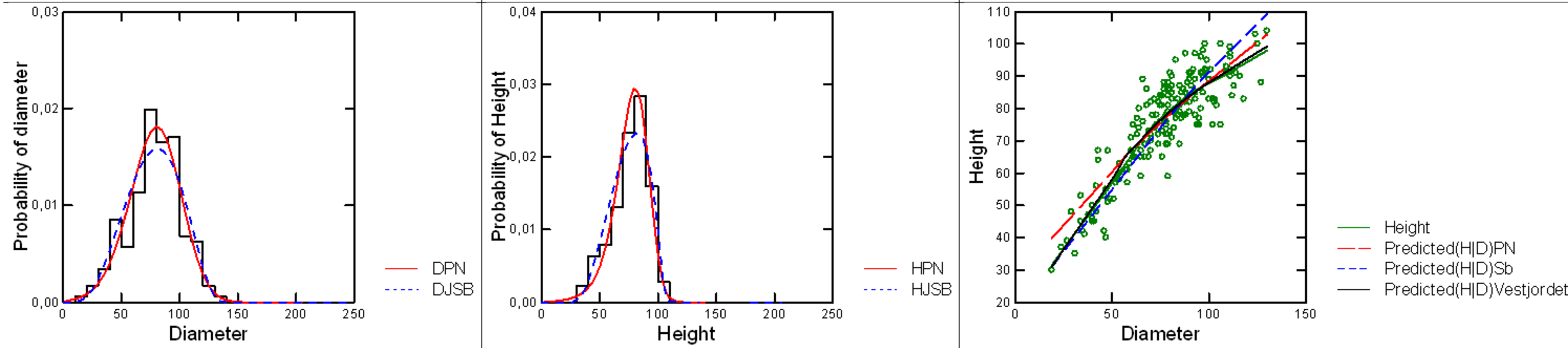




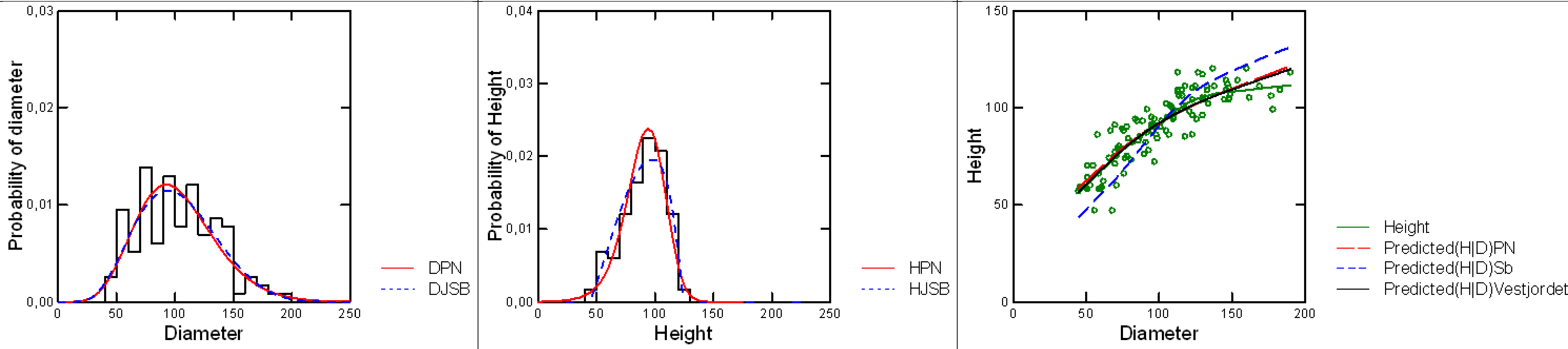
Results (↓) for STAND = 49



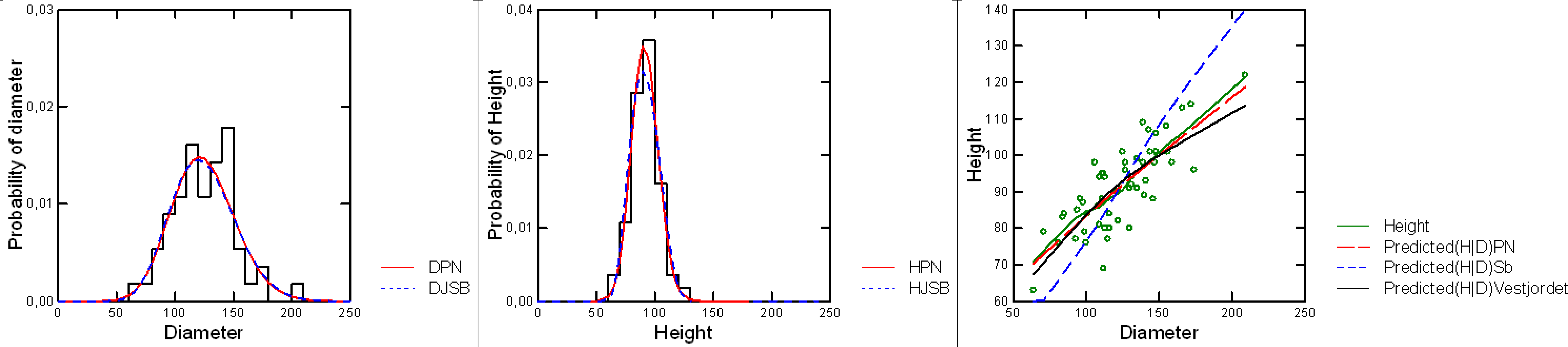
Results (↓) for STAND = 50



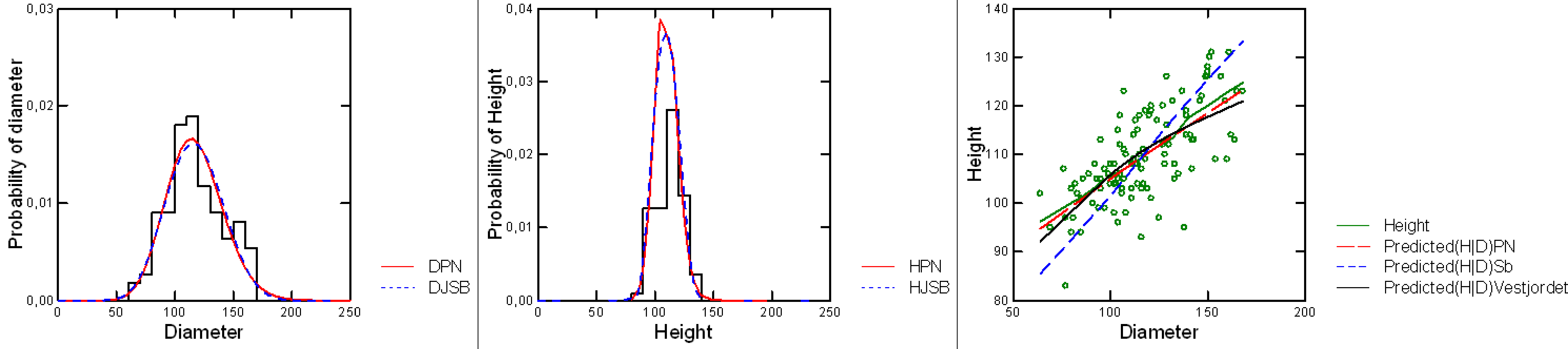
Results (↓) for STAND = 51



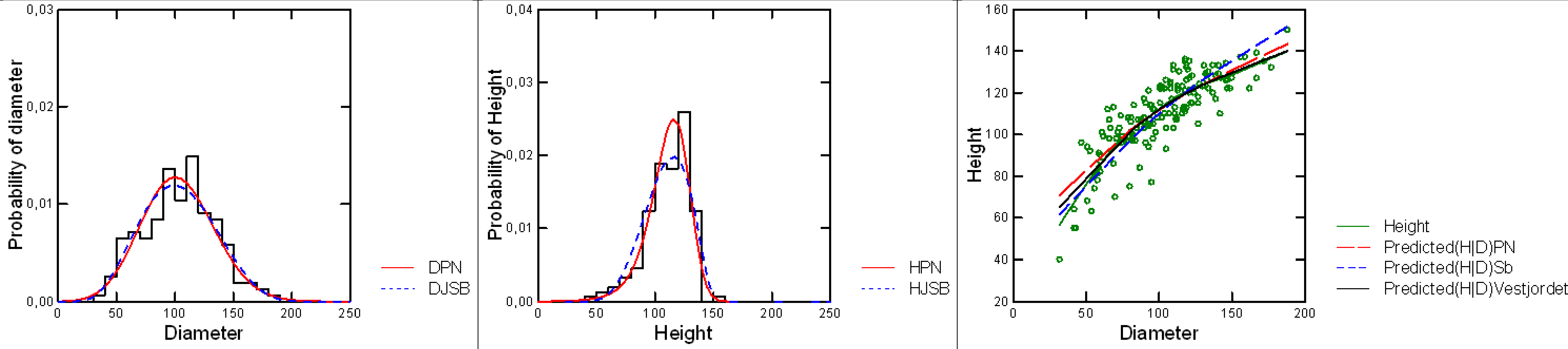
Results (↓) for STAND = 52



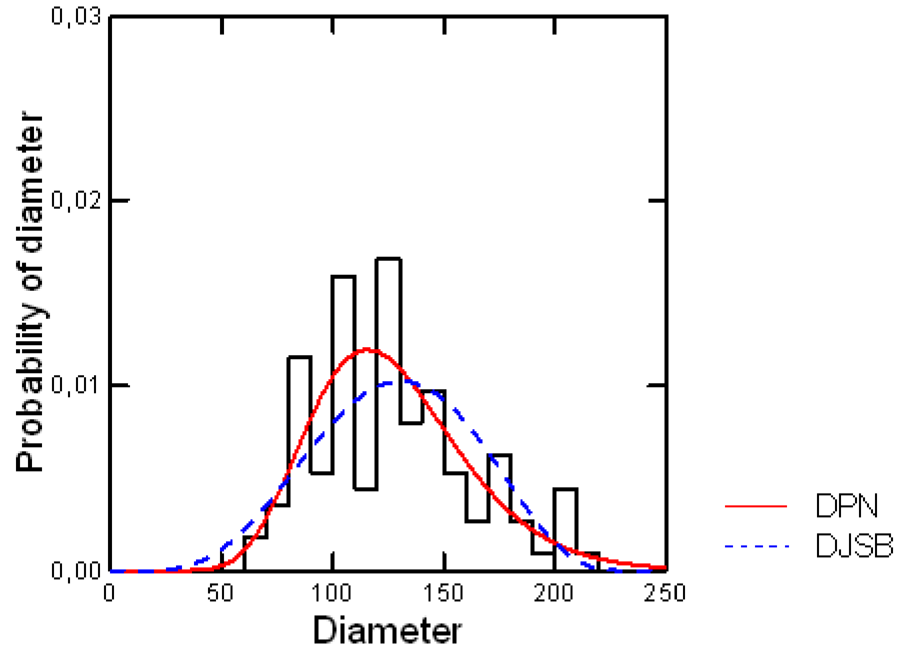
Results (↓) for STAND = 53

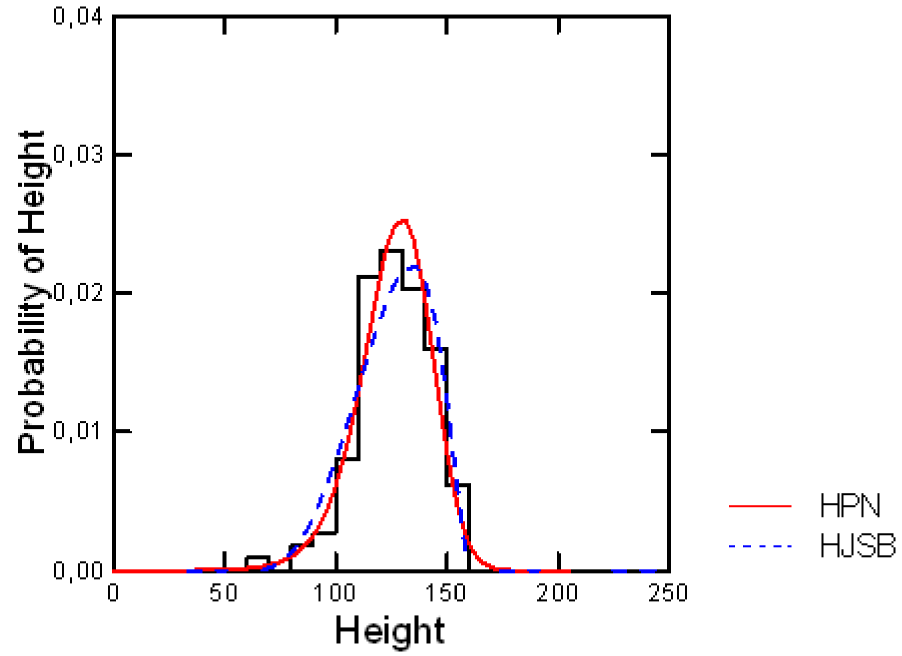


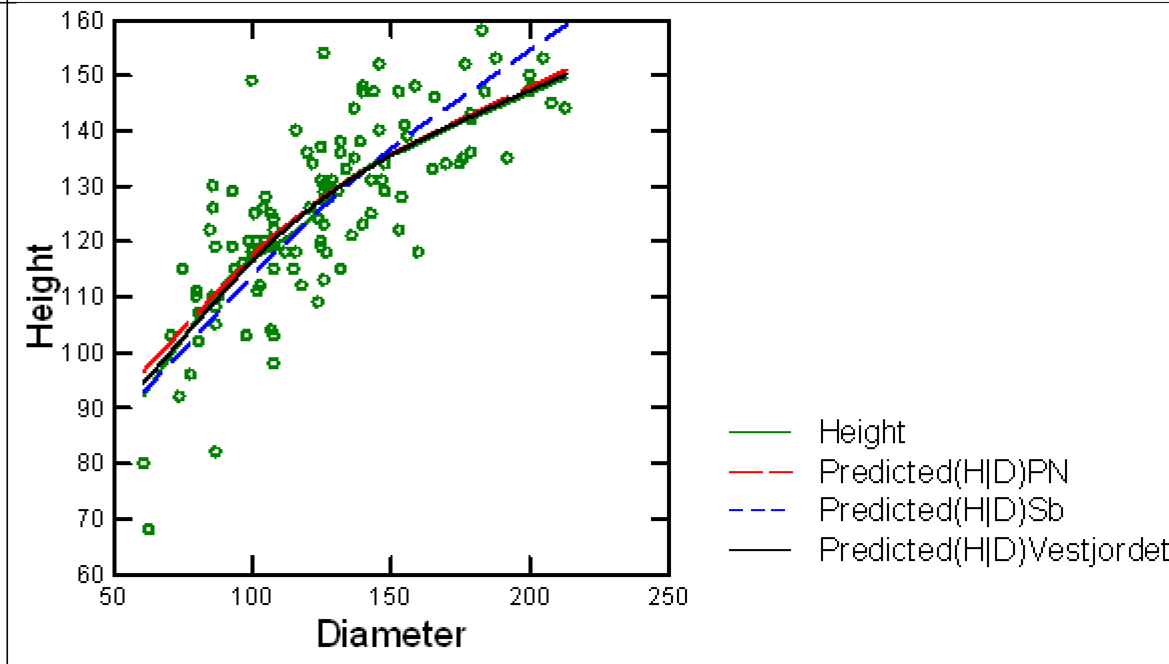
Results (↓) for STAND = 54



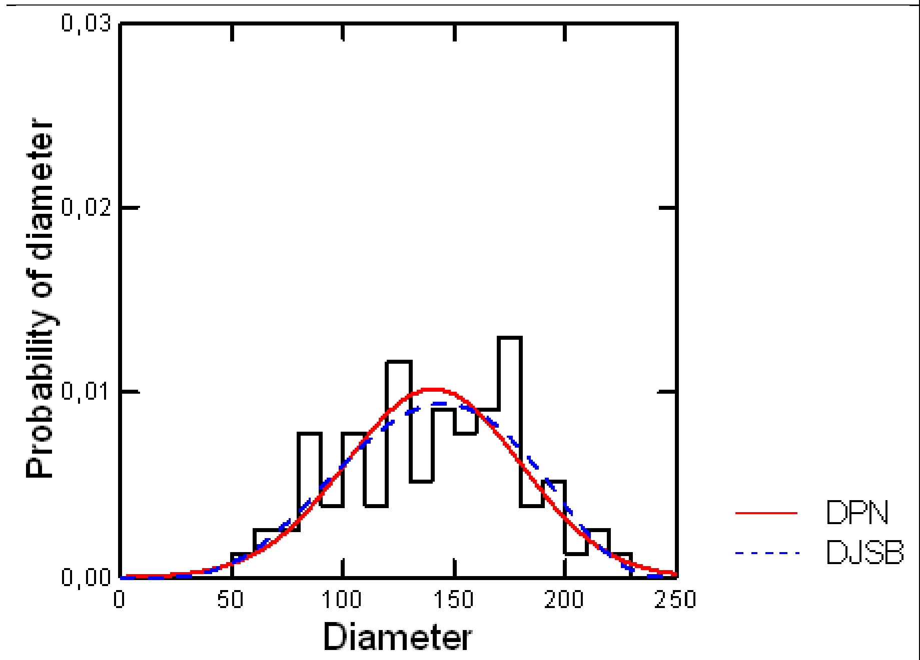
Results (↓) for STAND = 55

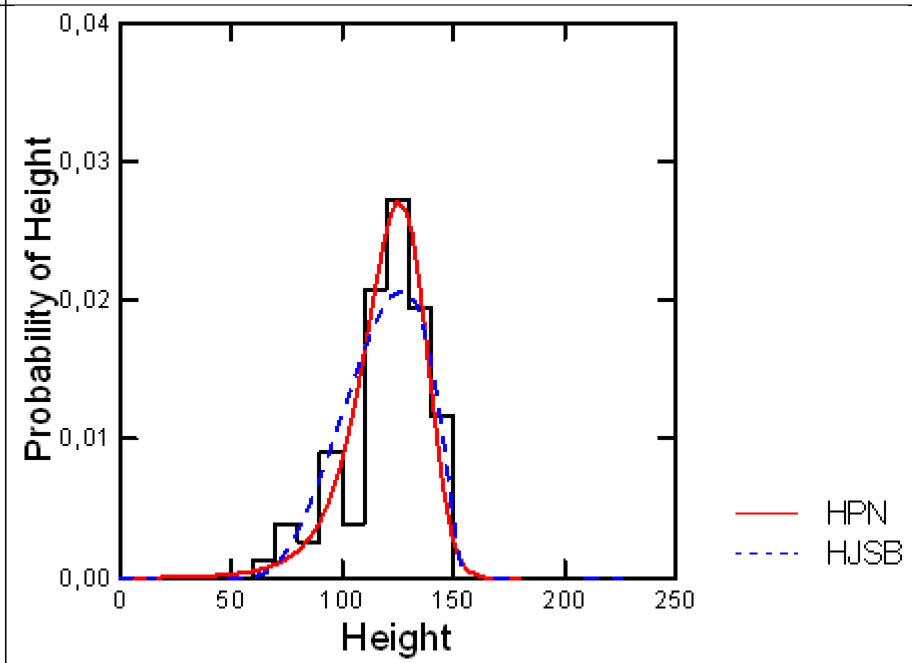


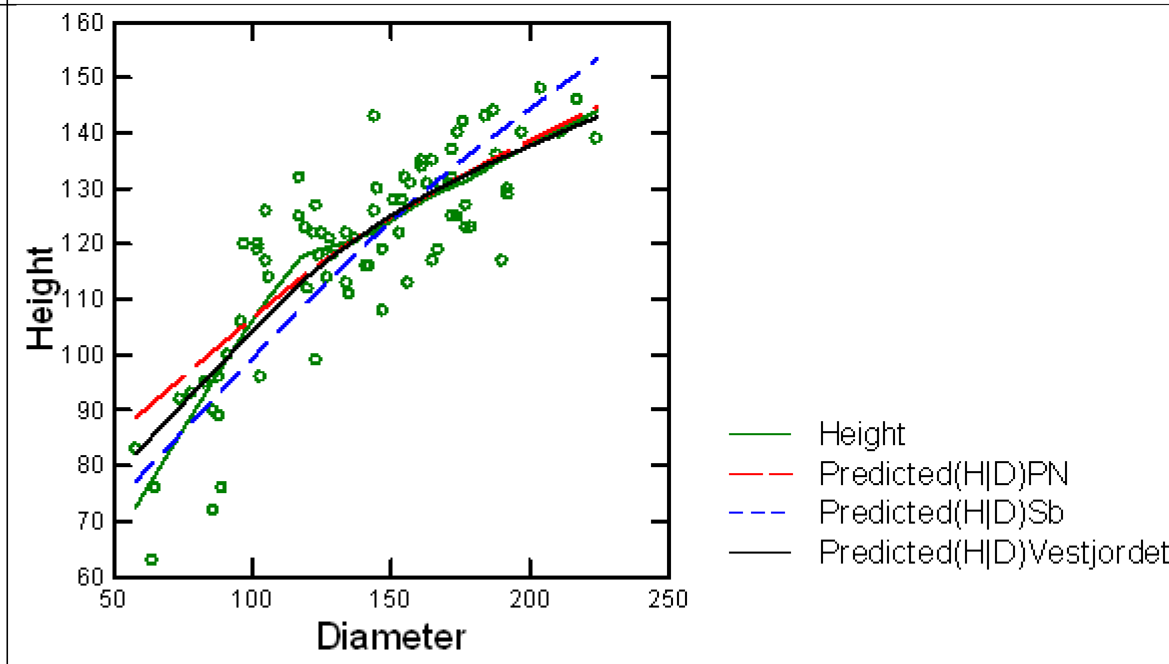




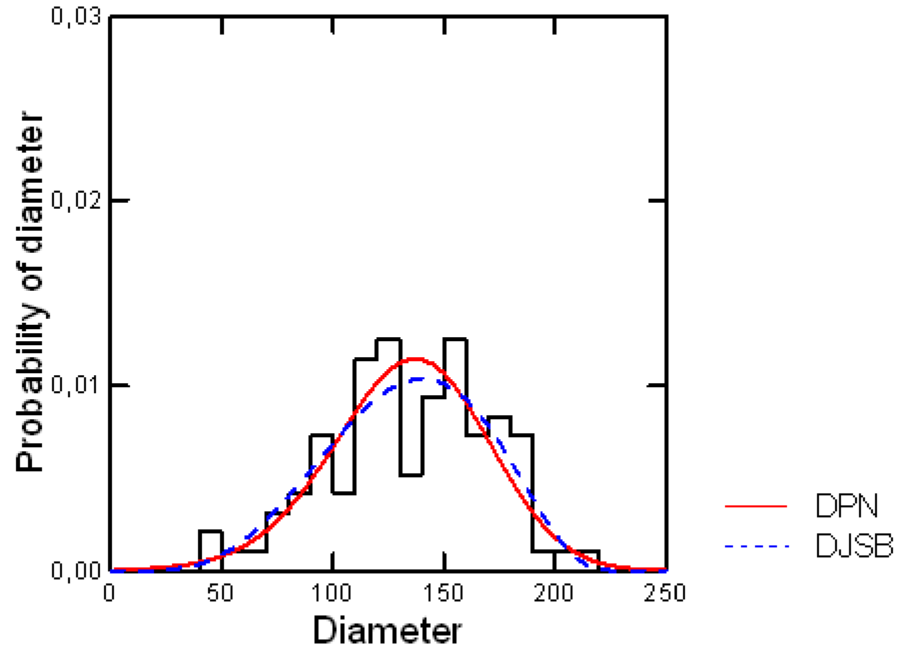
Results (↓) for STAND = 56

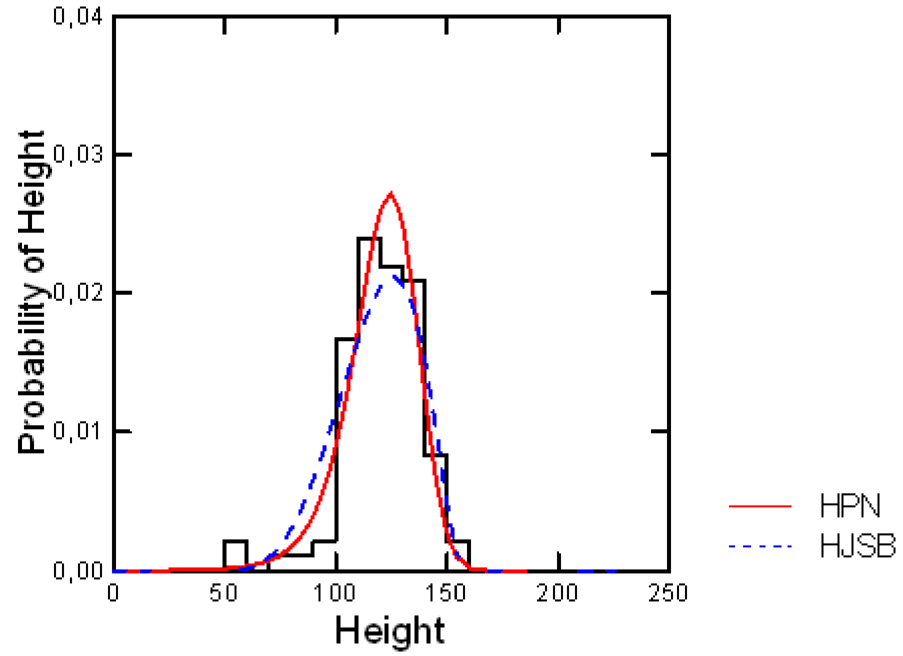


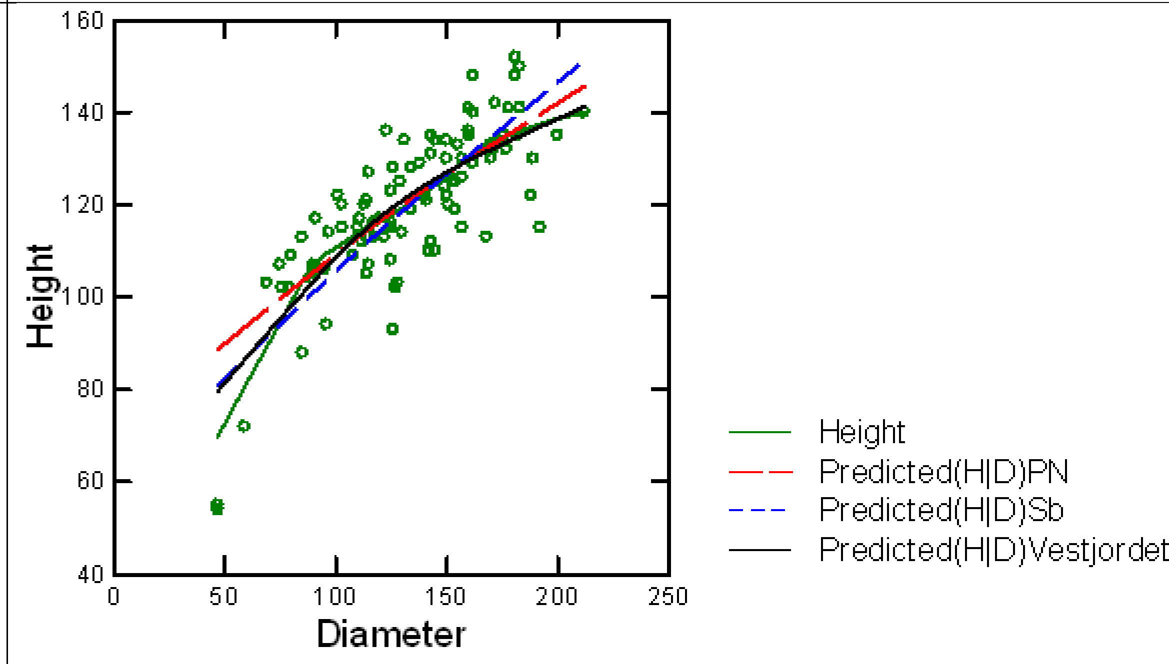




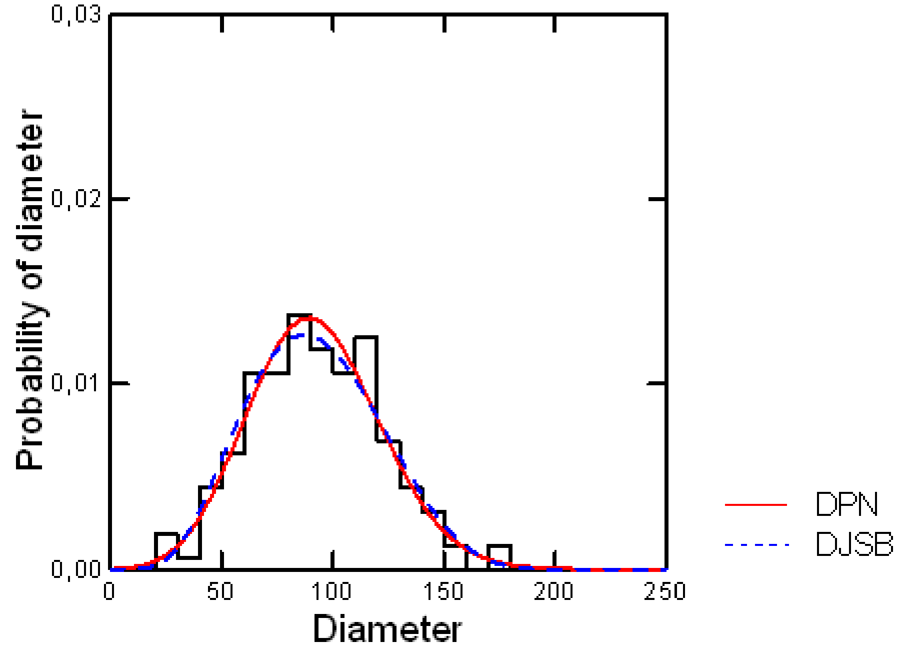
Results (↓) for STAND = 57

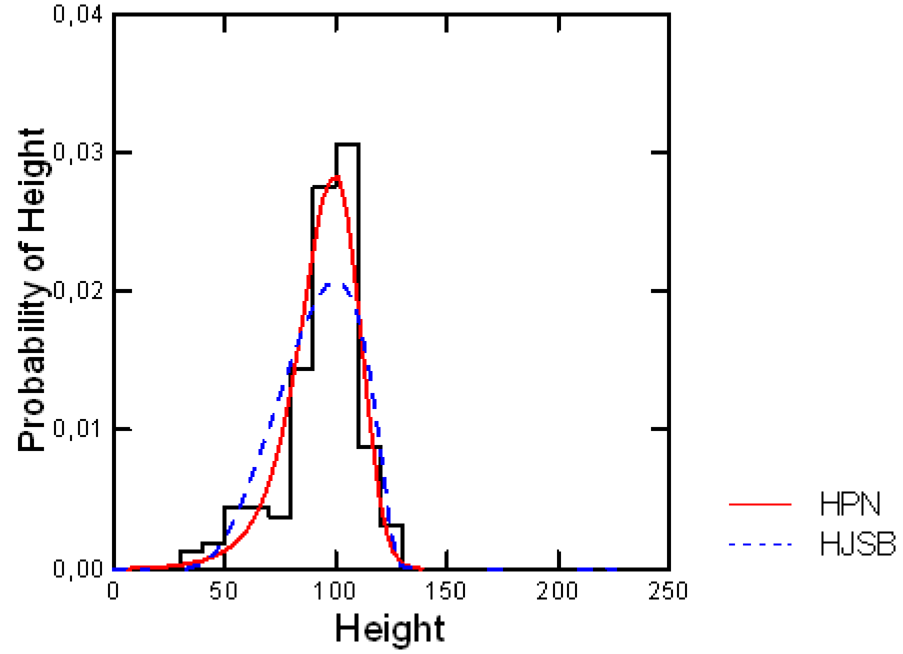


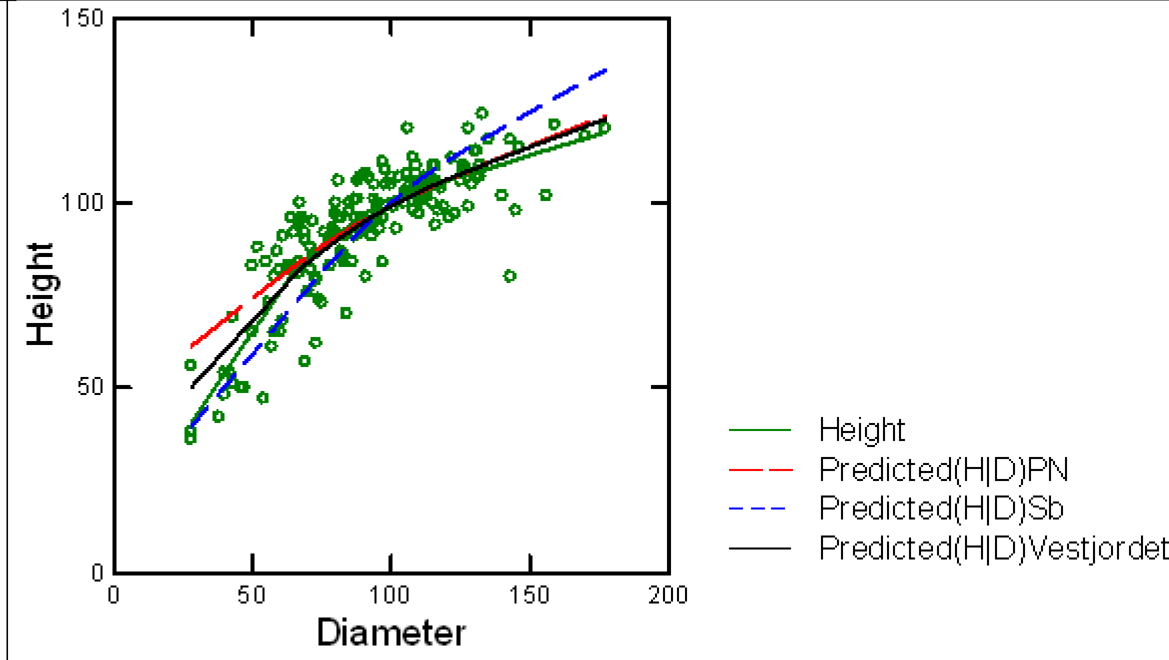




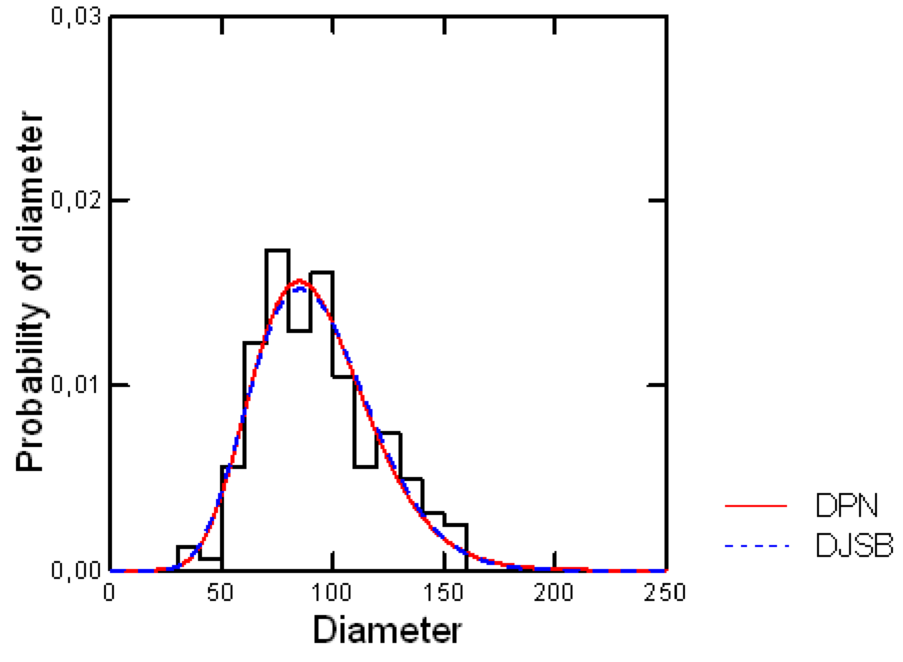
Results (↓) for STAND = 58

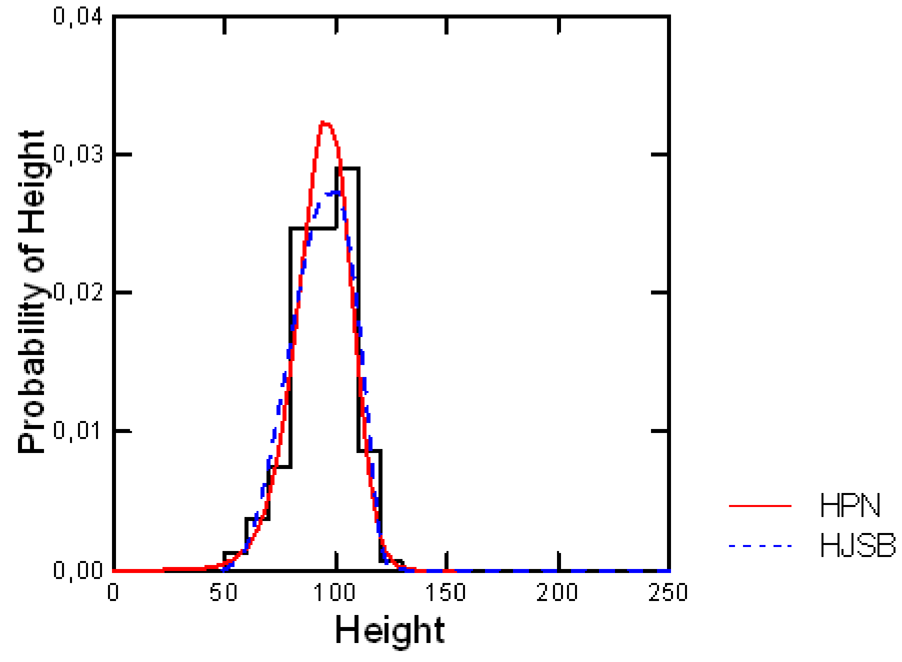


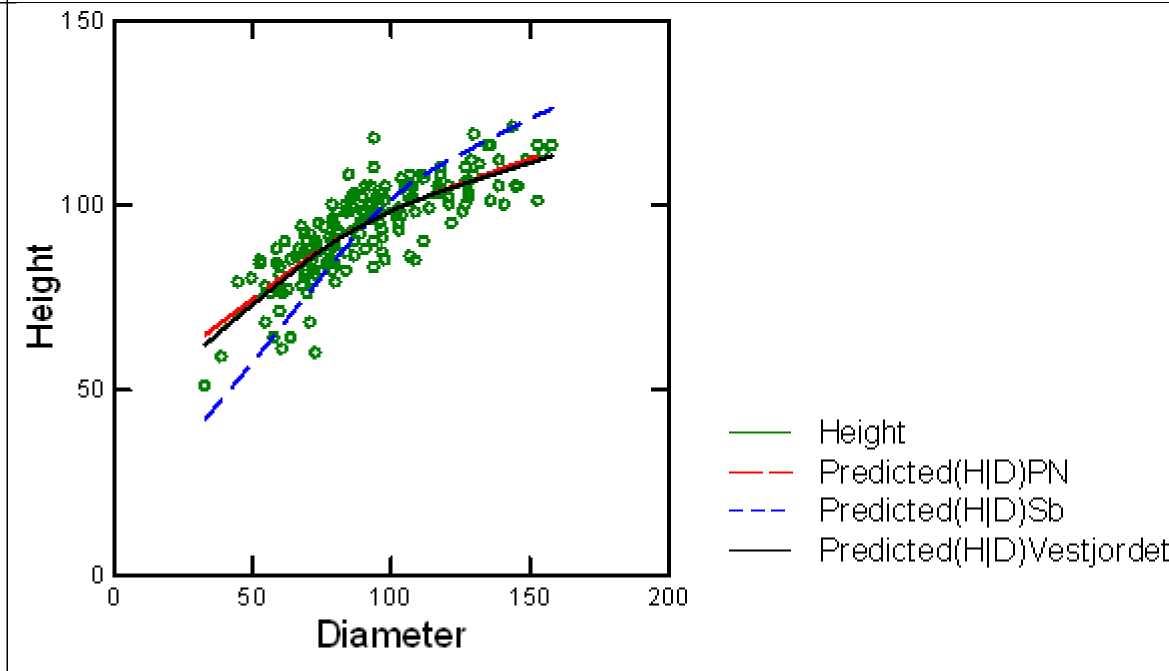




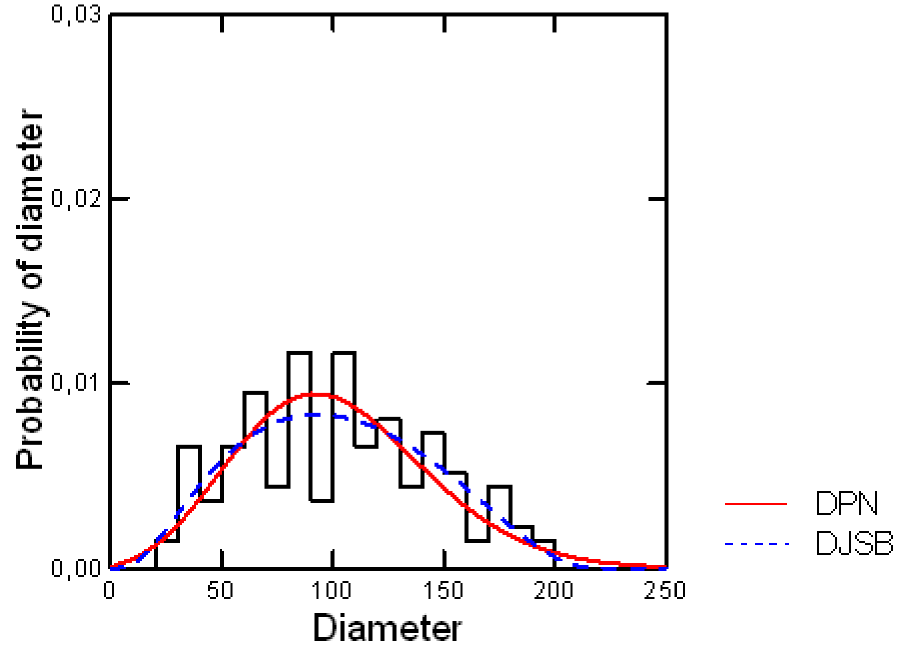
Results (↓) for STAND = 59

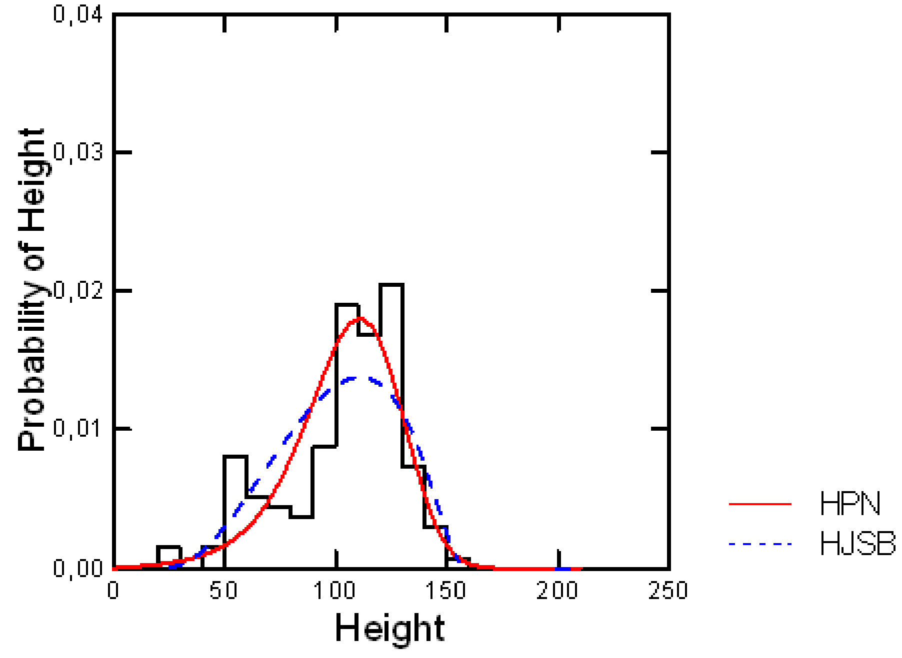


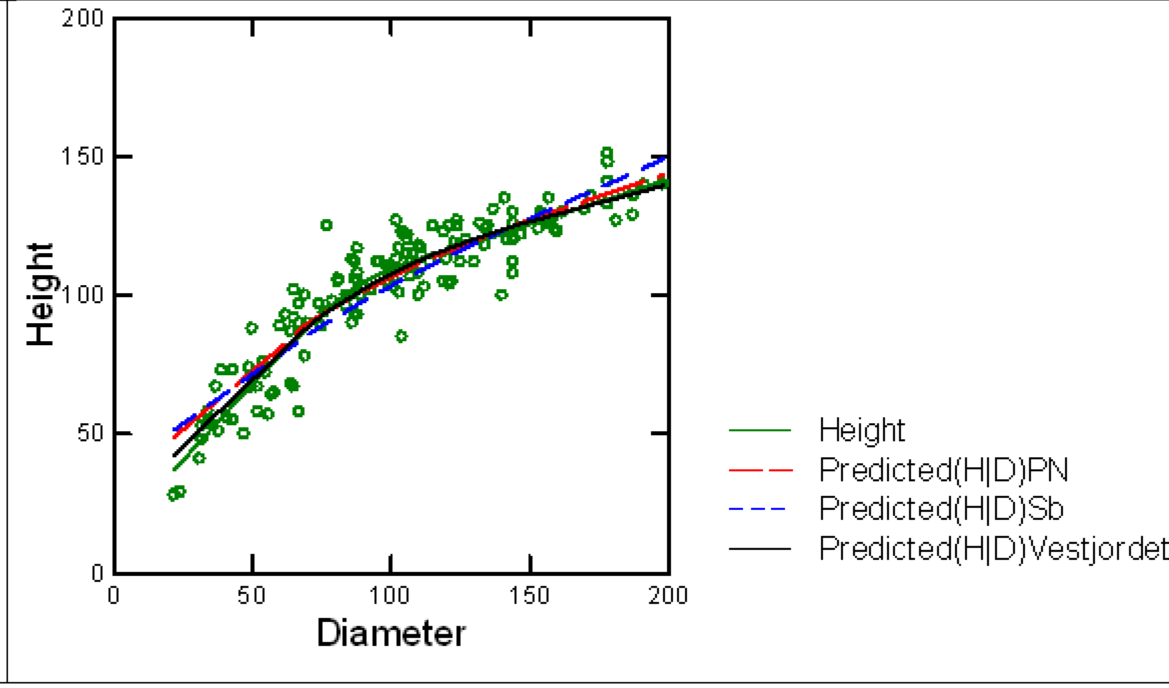




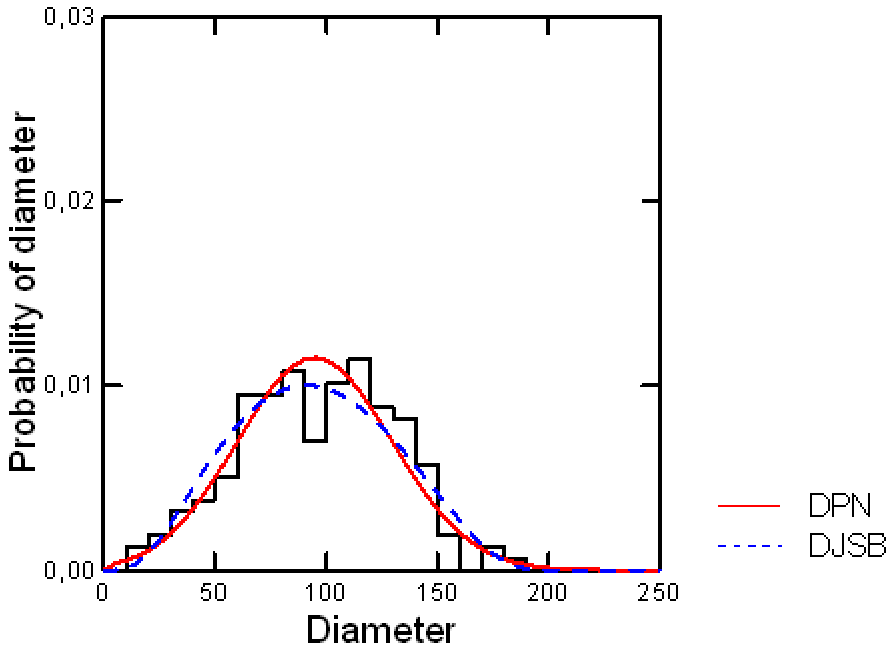
Results (↓) for STAND = 60

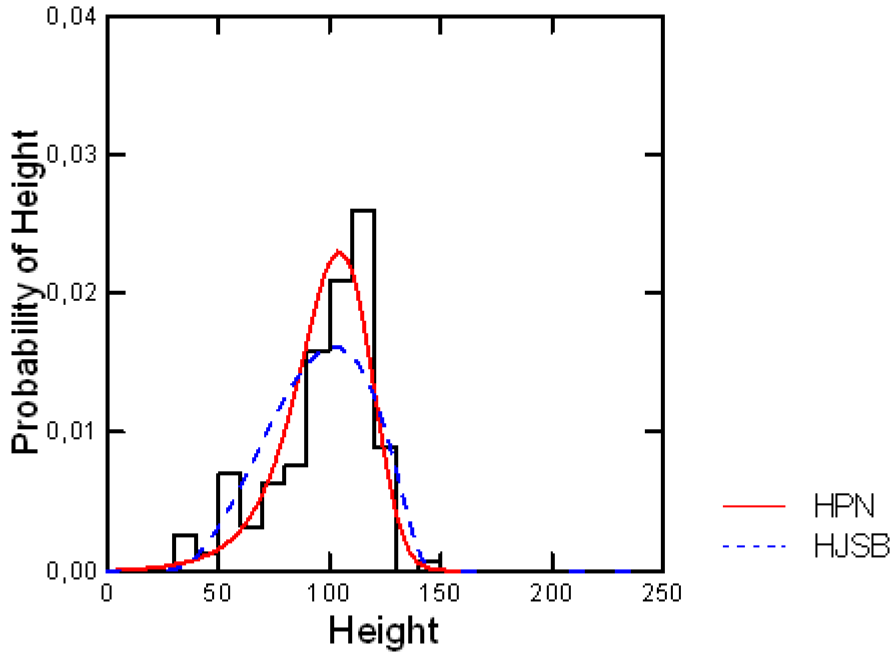


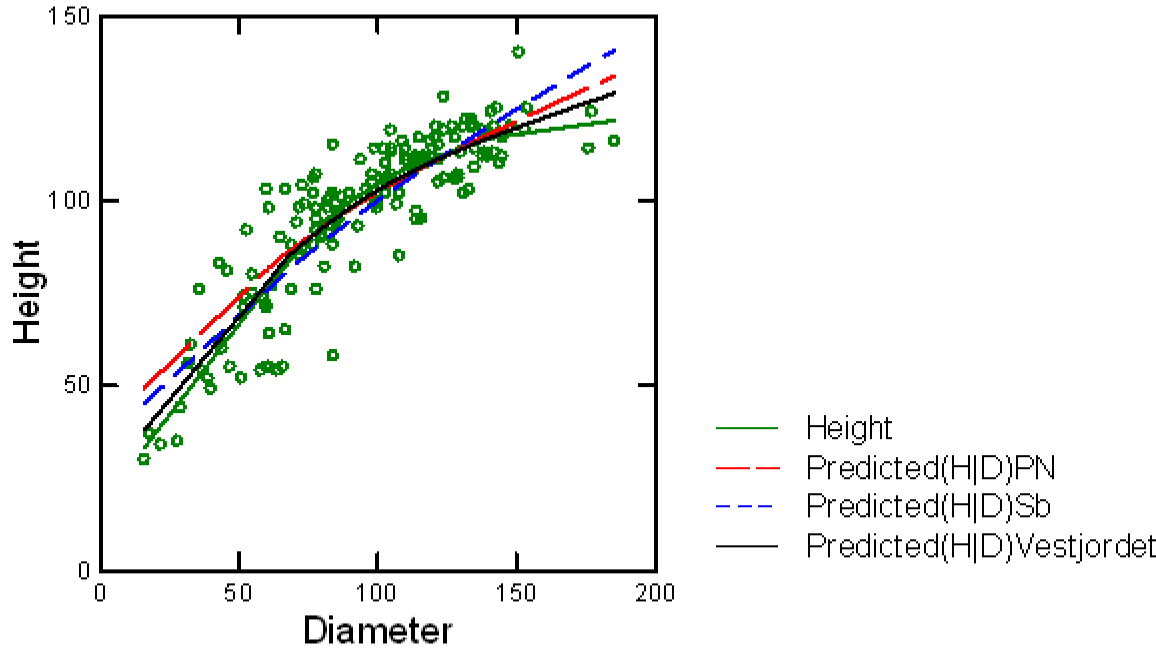




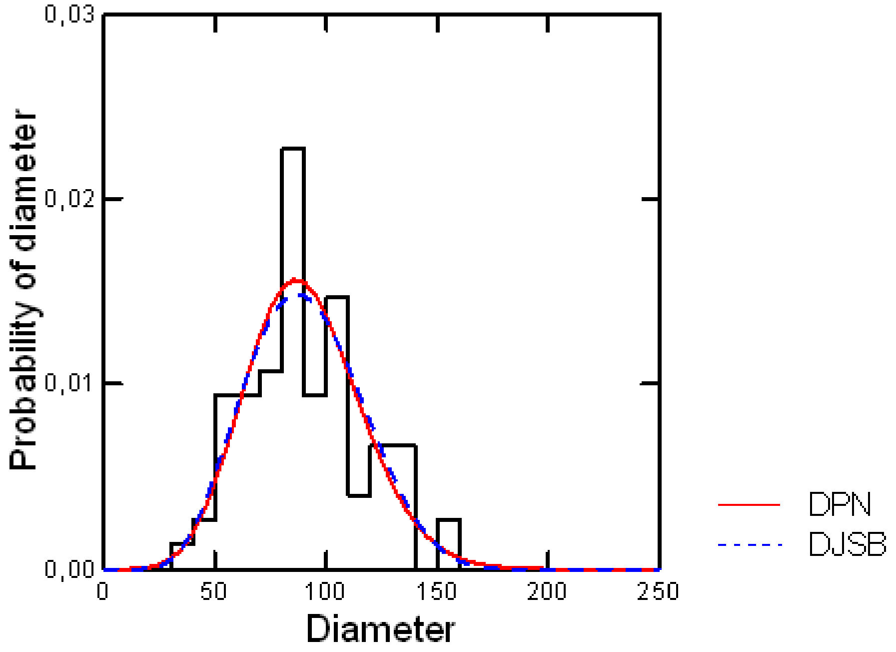
Results (↓) for STAND = 61

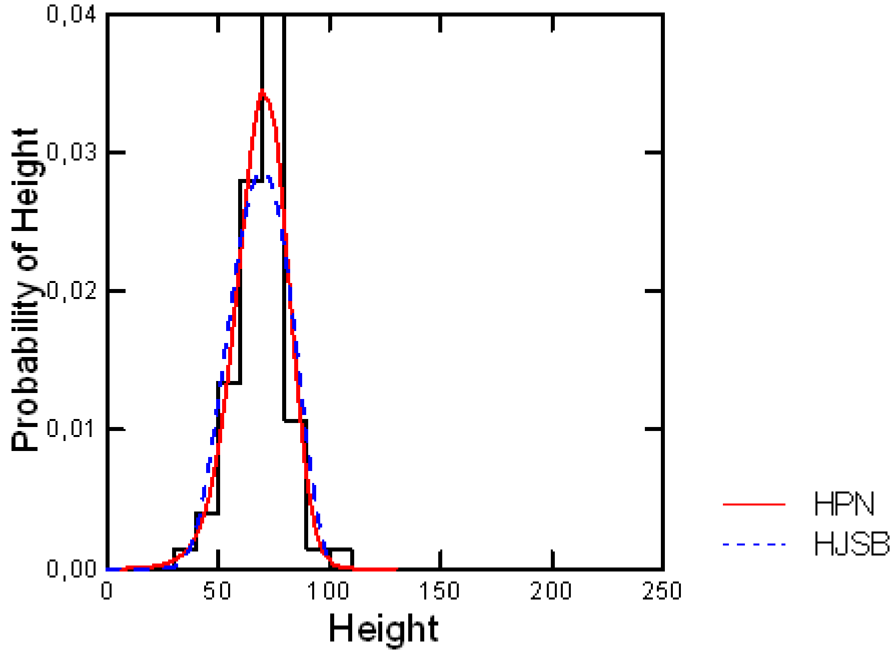


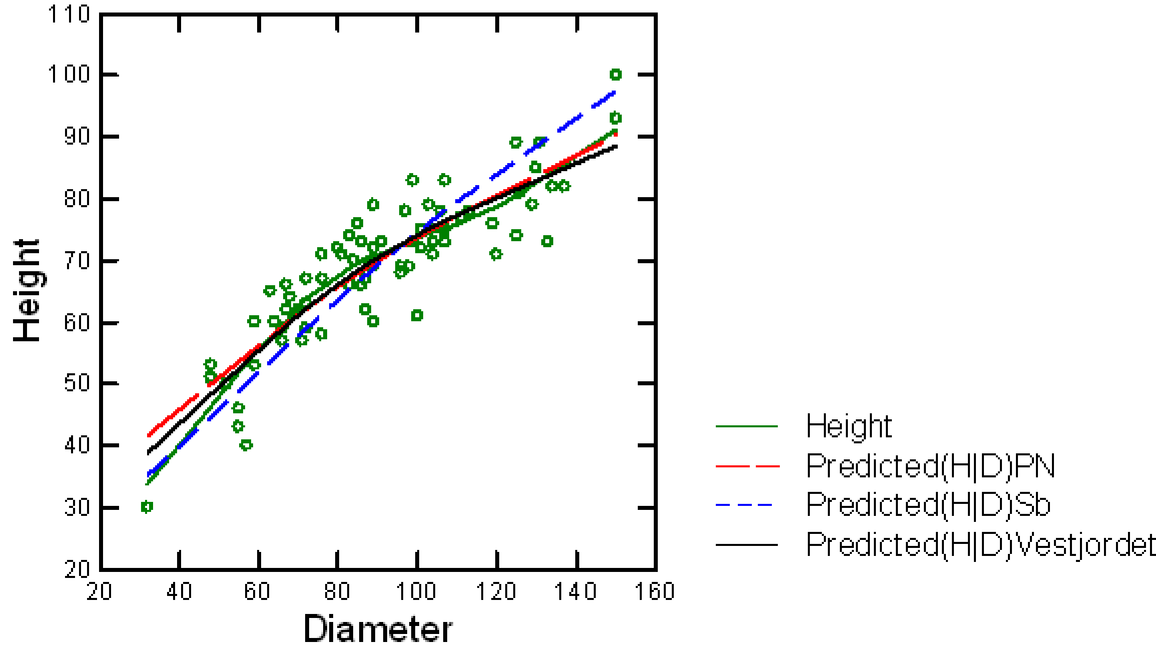




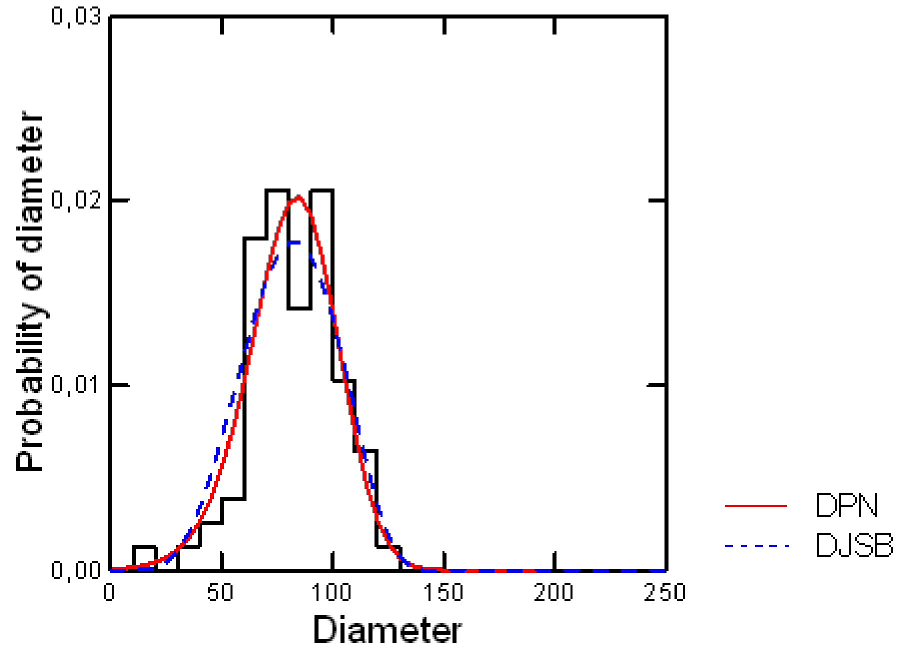
Results (↓) for STAND = 62

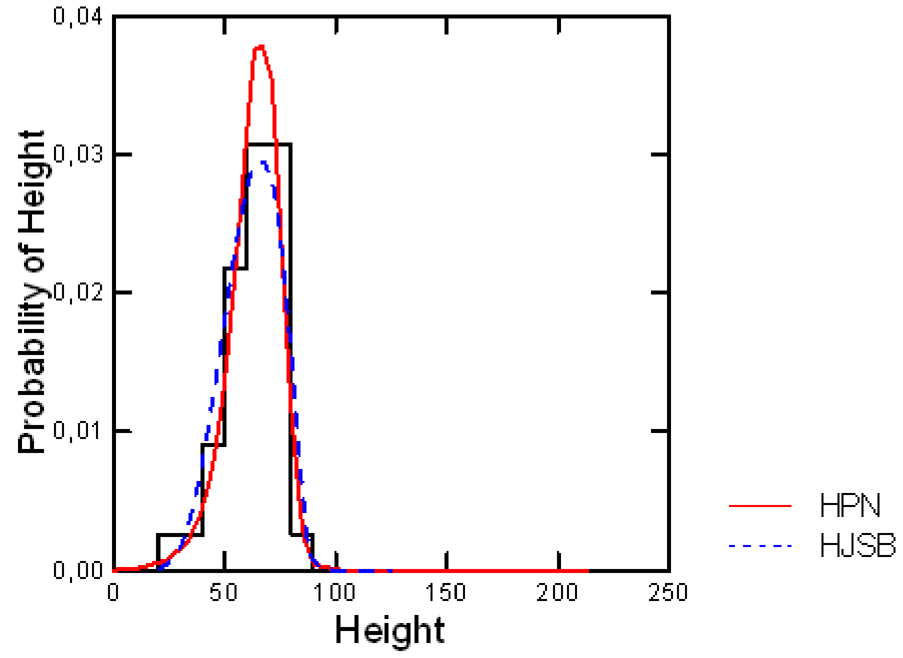


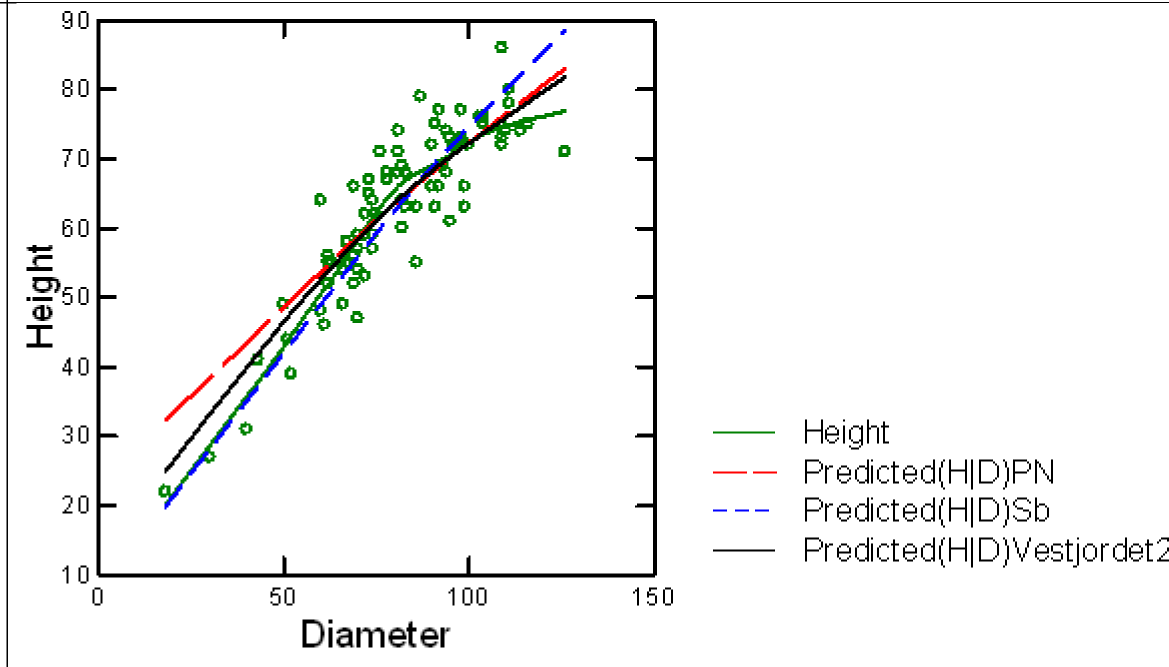




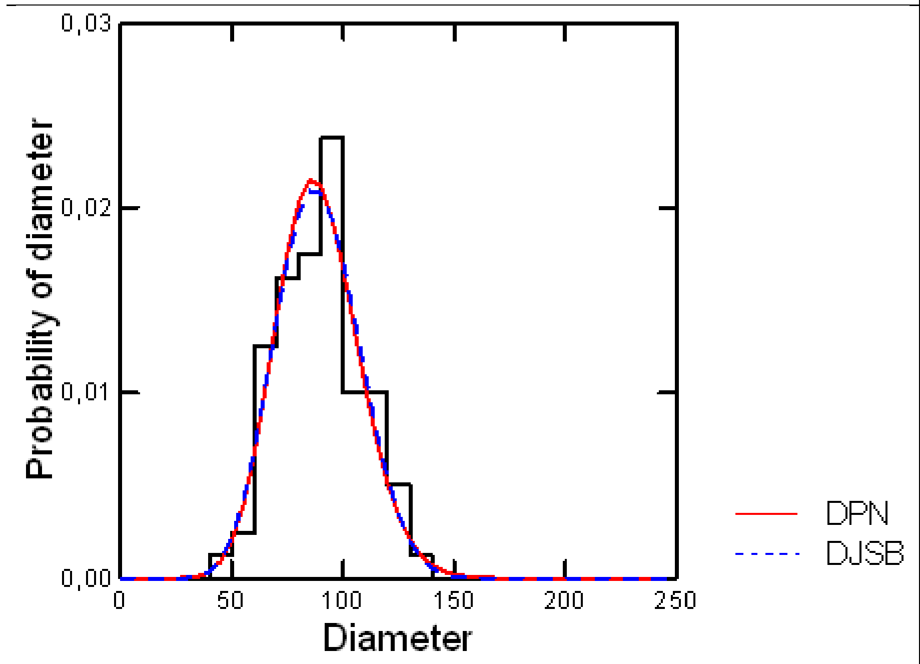
Results (↓) for STAND = 63

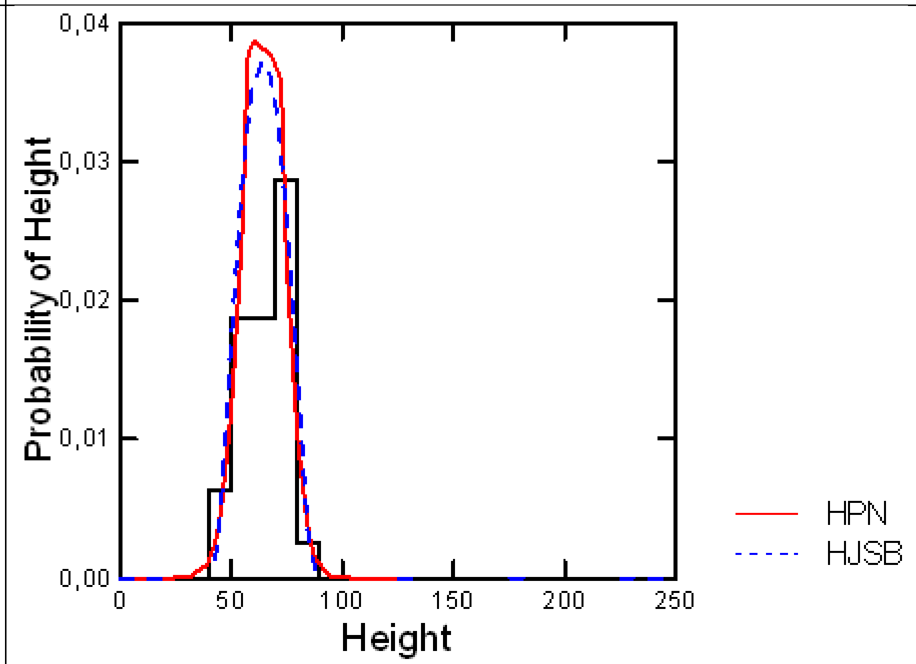


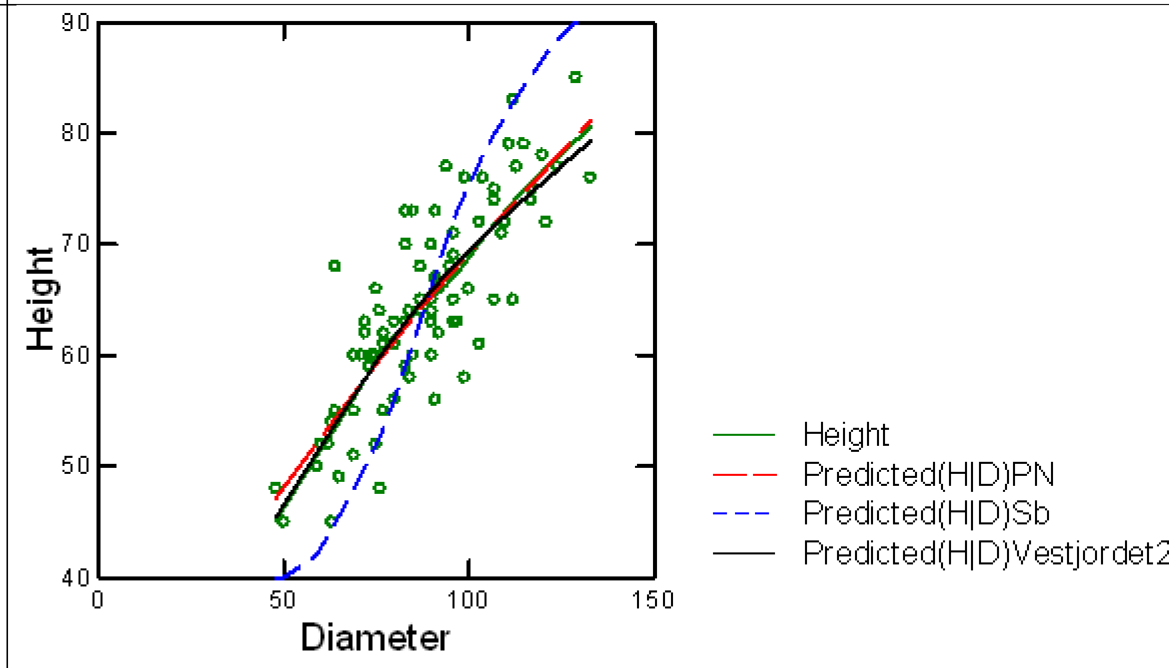




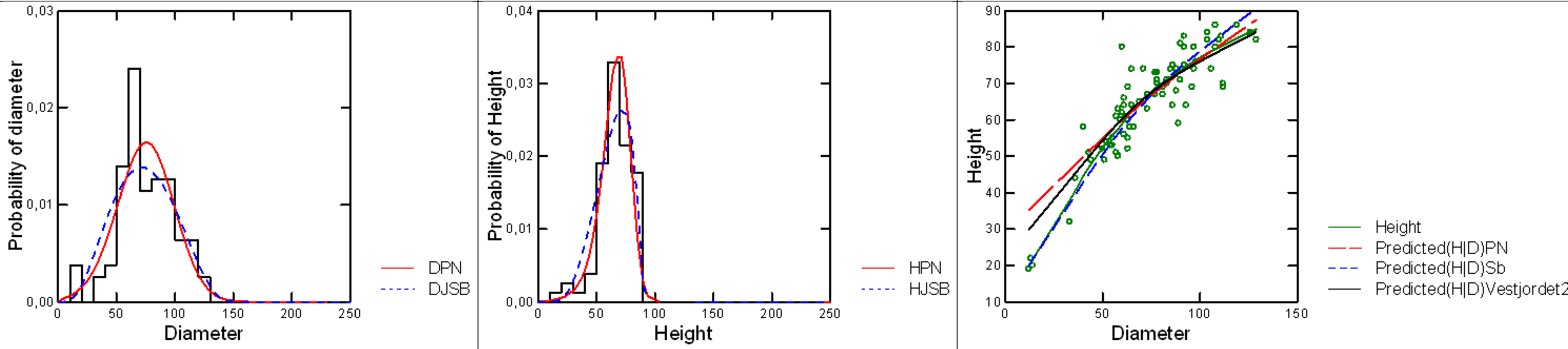
Results (↓) for STAND = 64



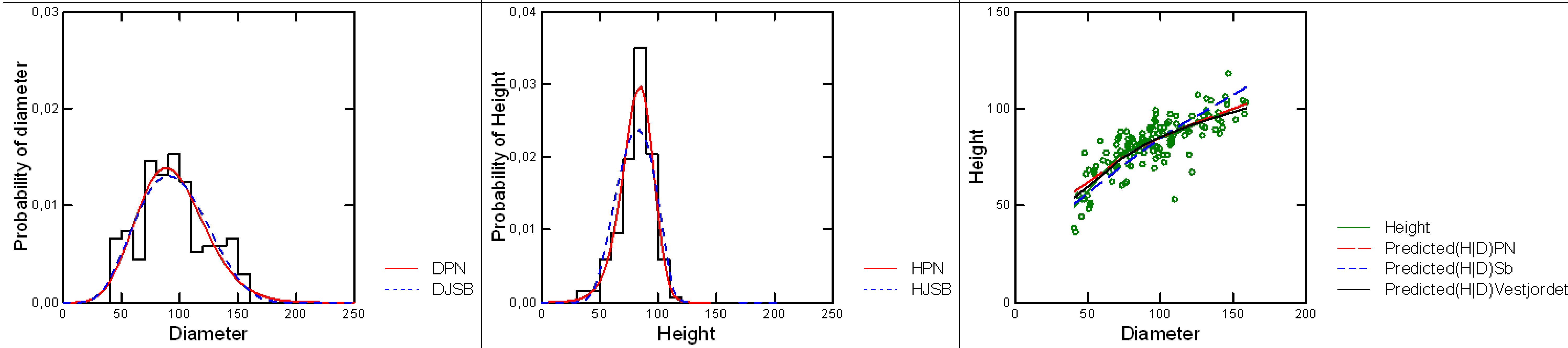




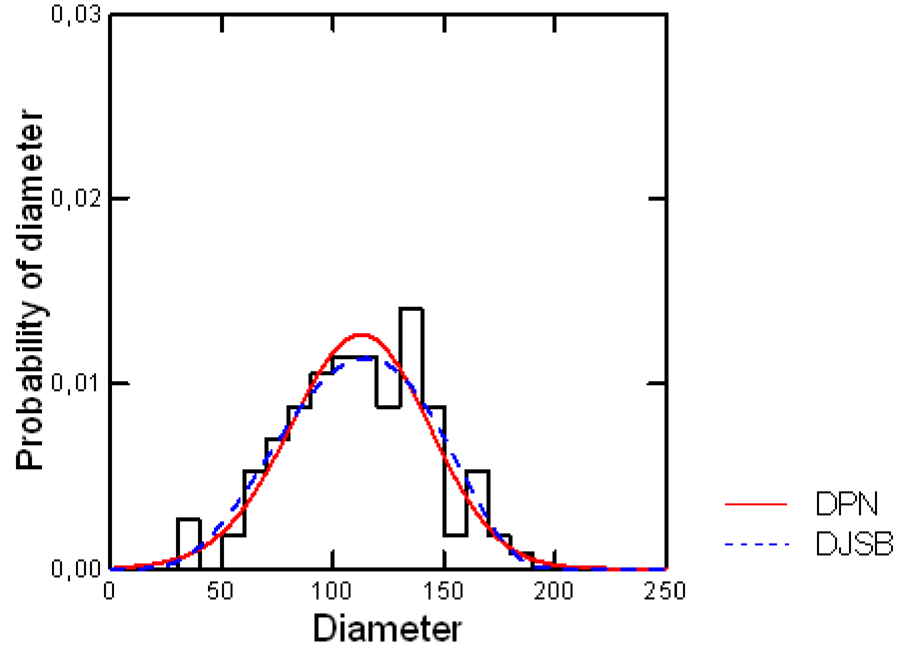
Results (↓) for STAND = 65

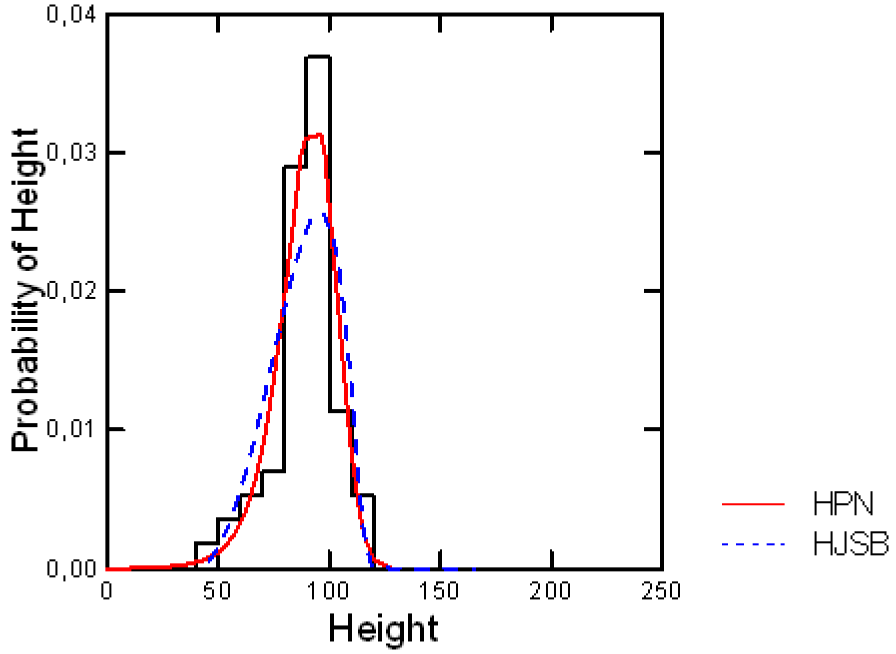


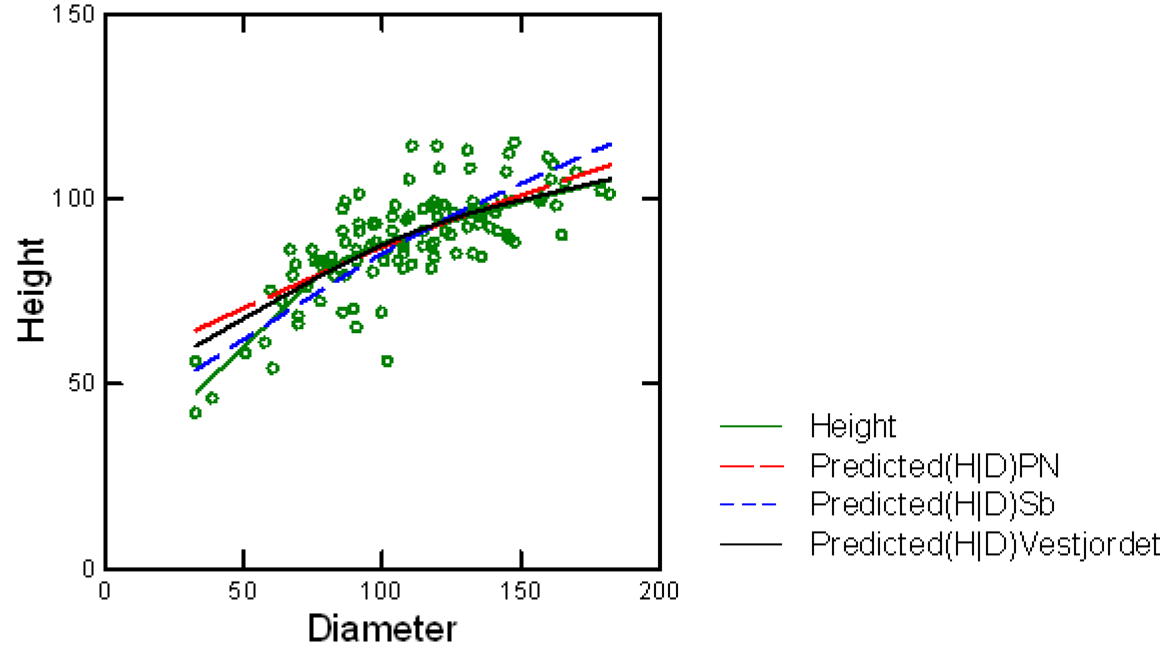
Results (↓) for STAND = 66



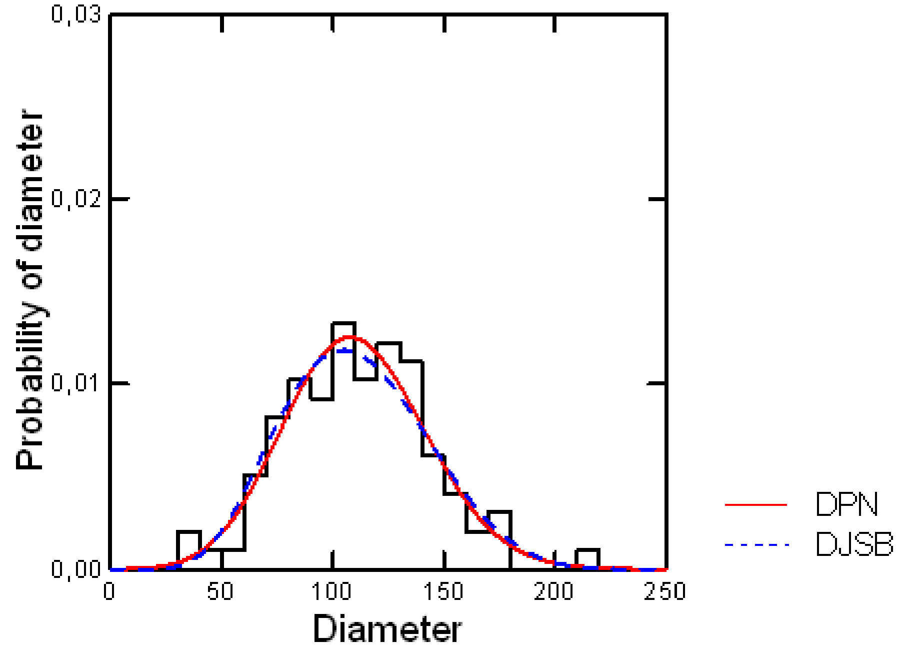
Results (↓) for STAND = 67

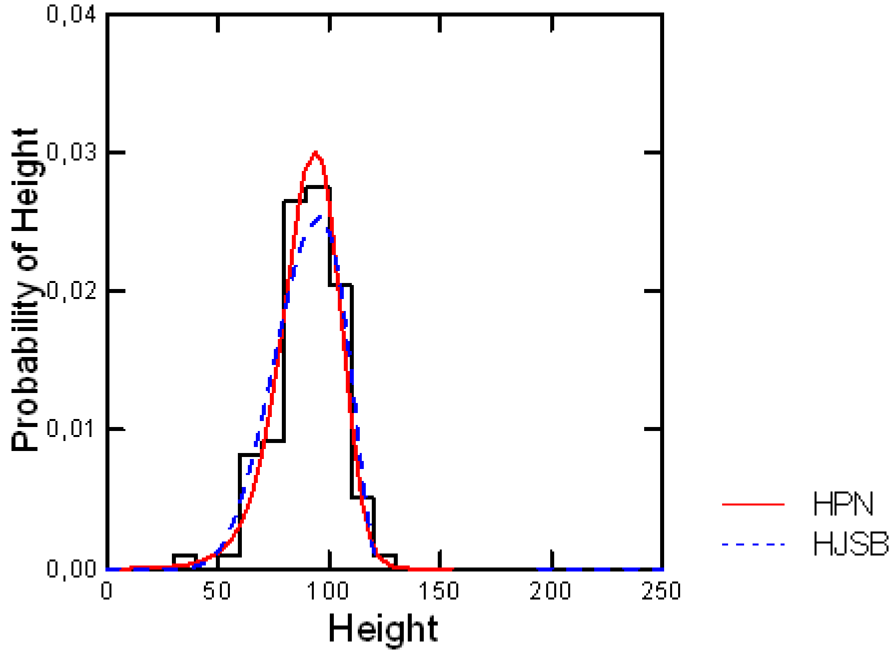


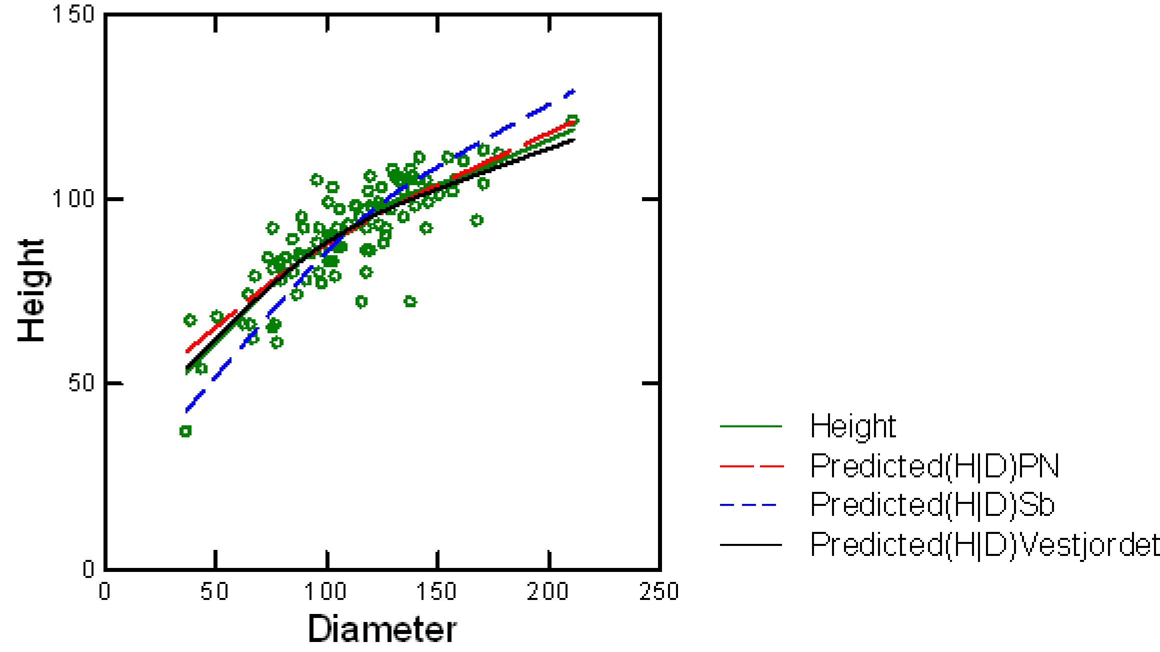




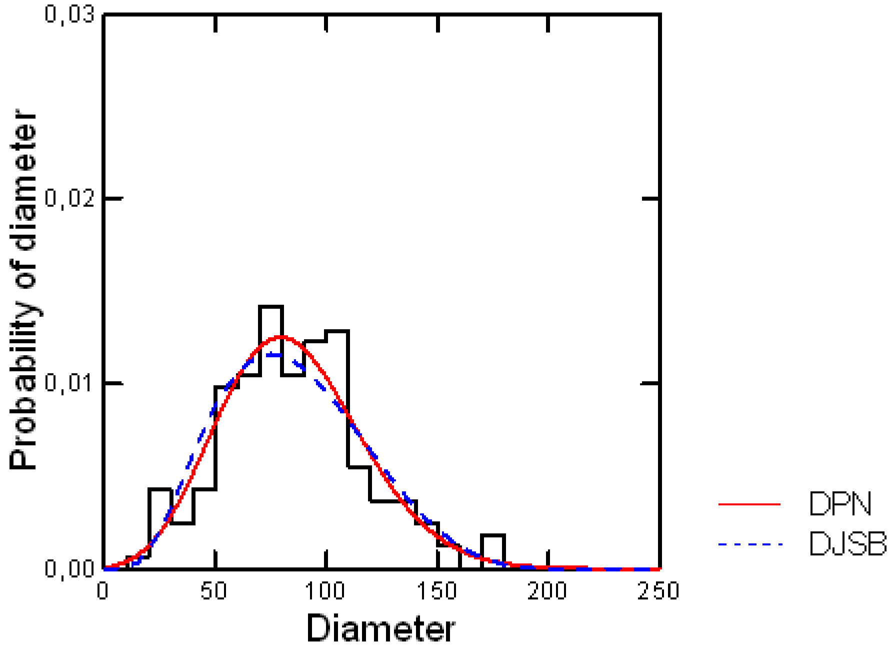
Results (↓) for STAND = 68

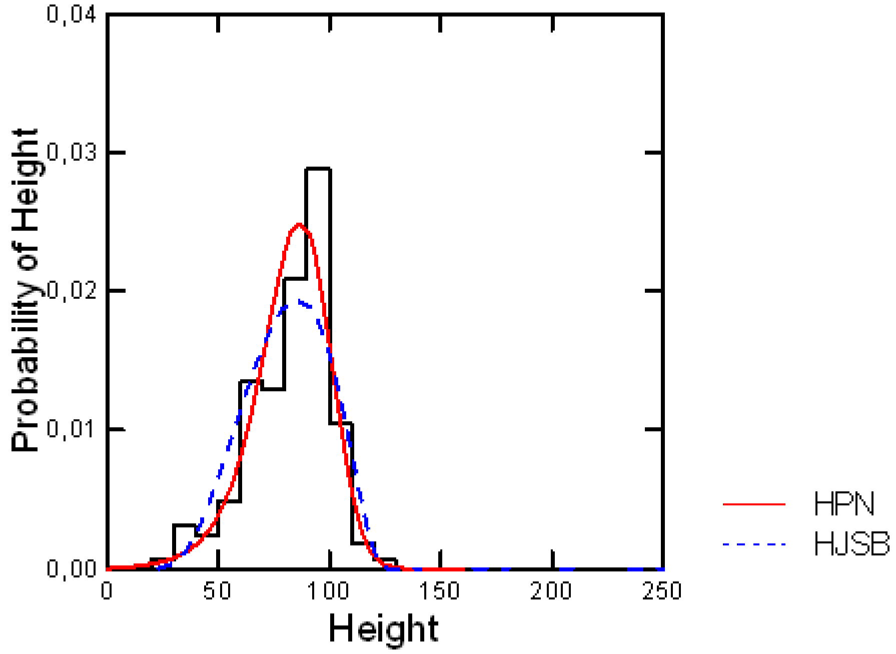


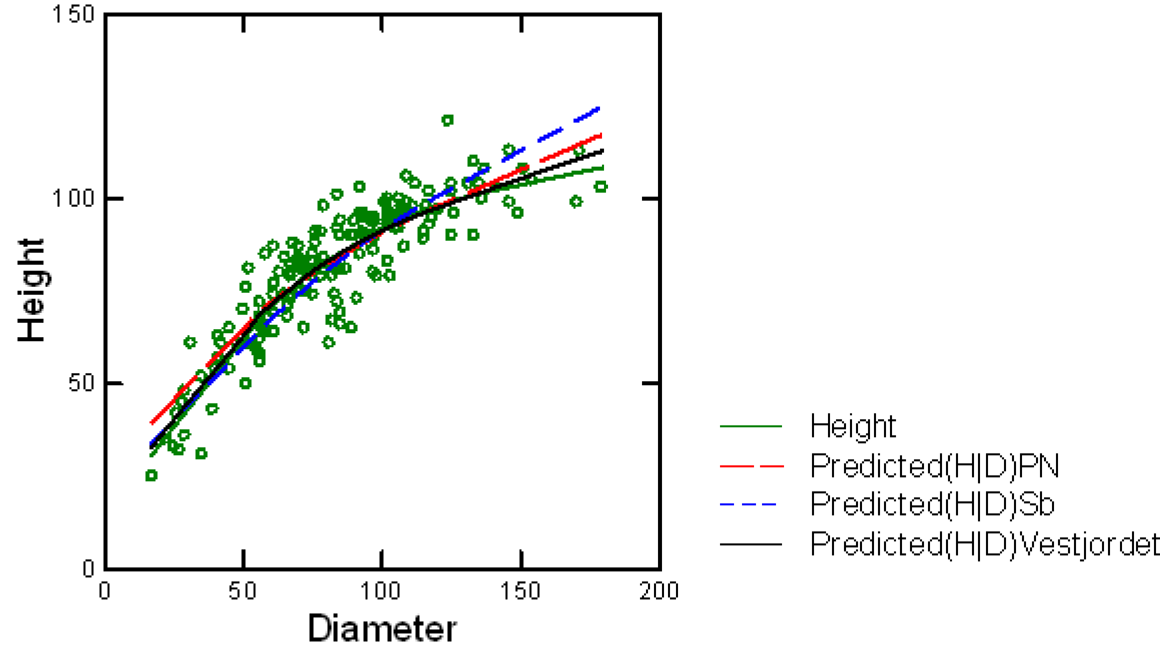




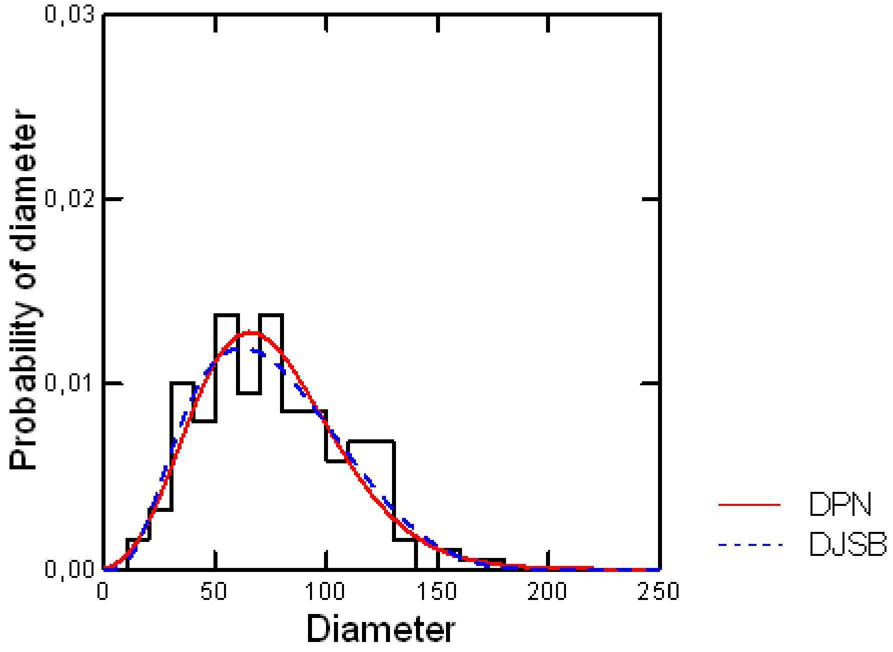
Results (↓) for STAND = 69

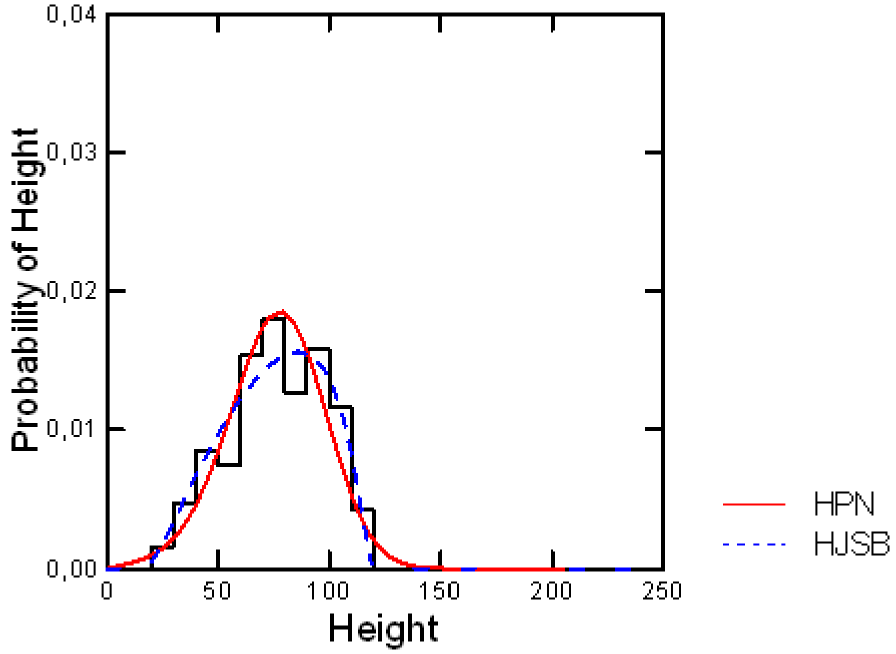


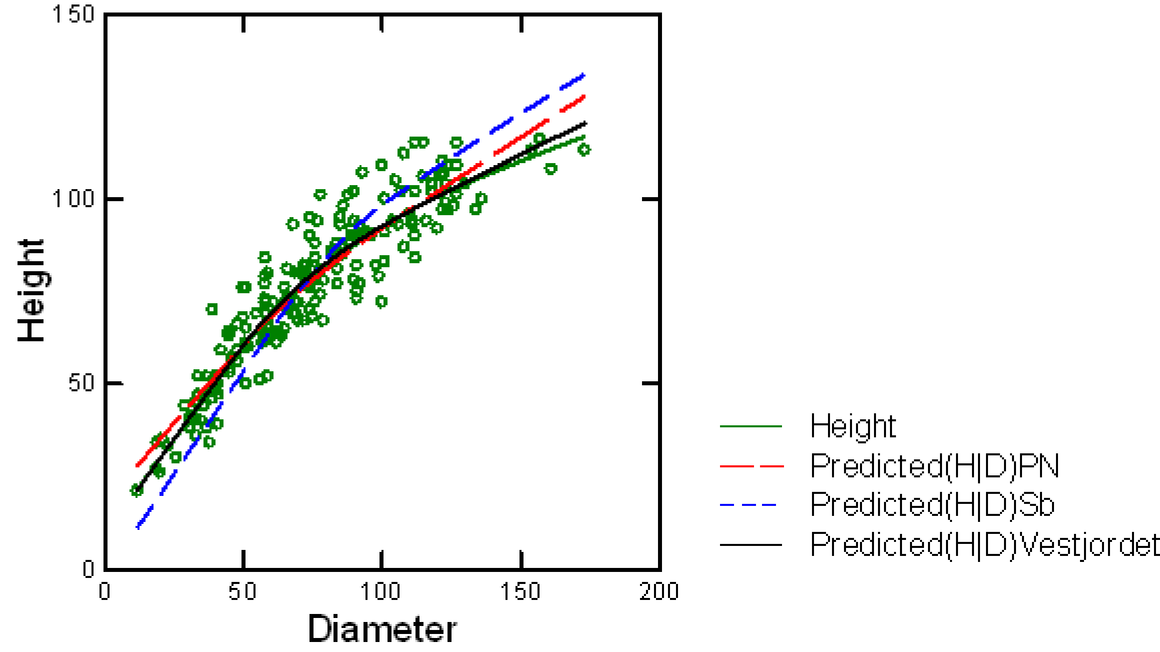




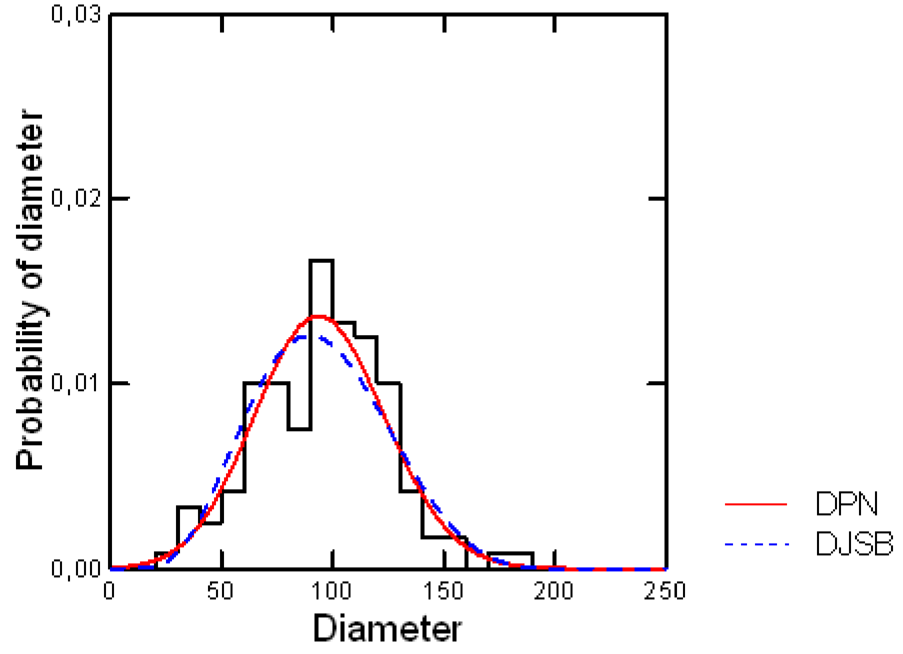
Results (↓) for STAND = 70

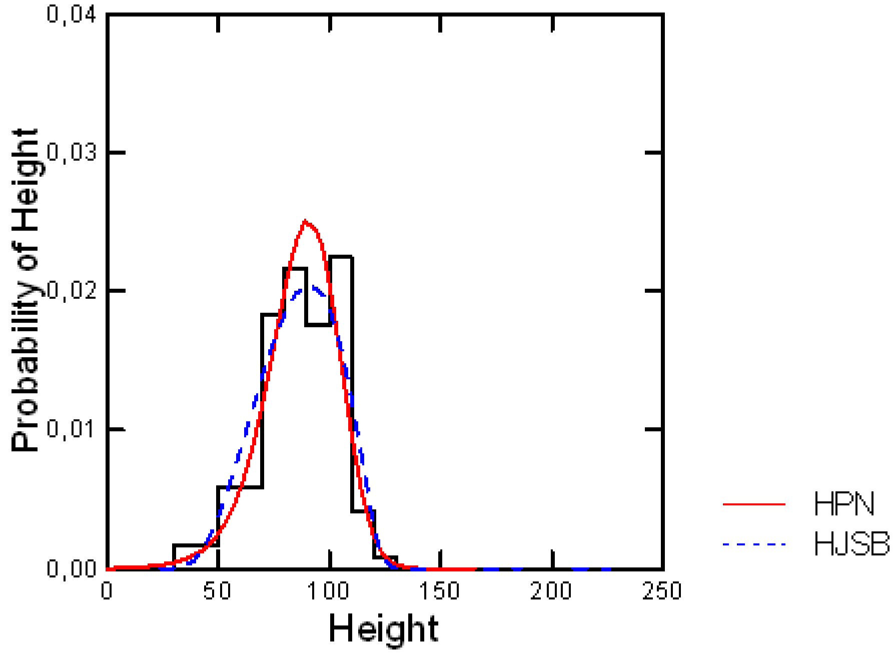


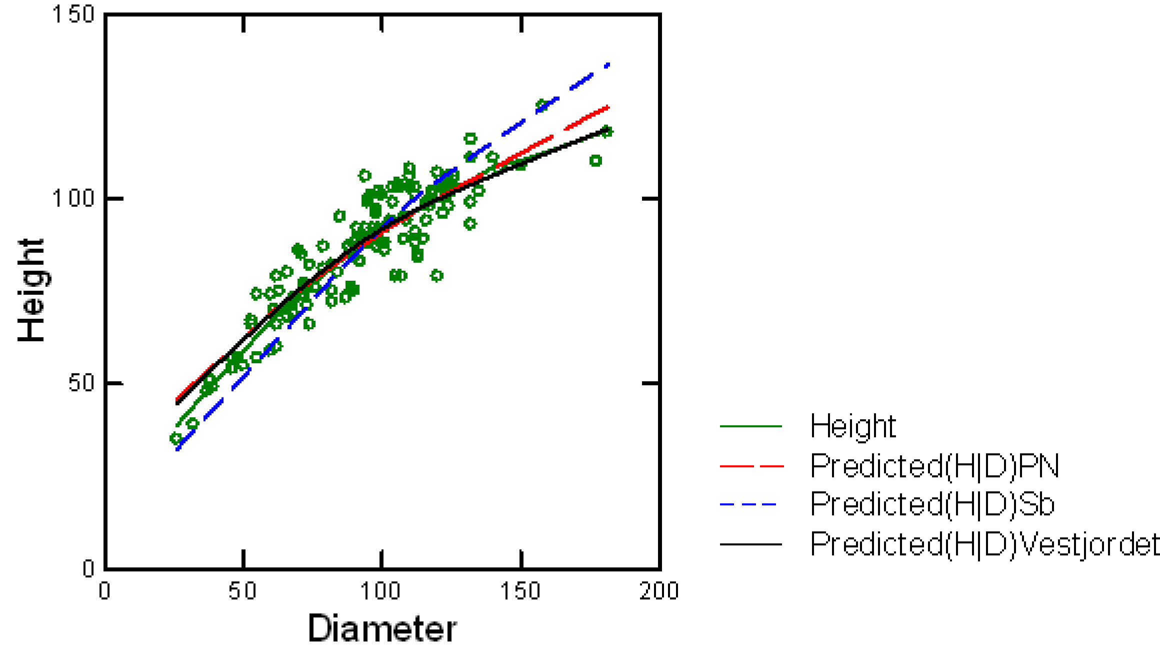




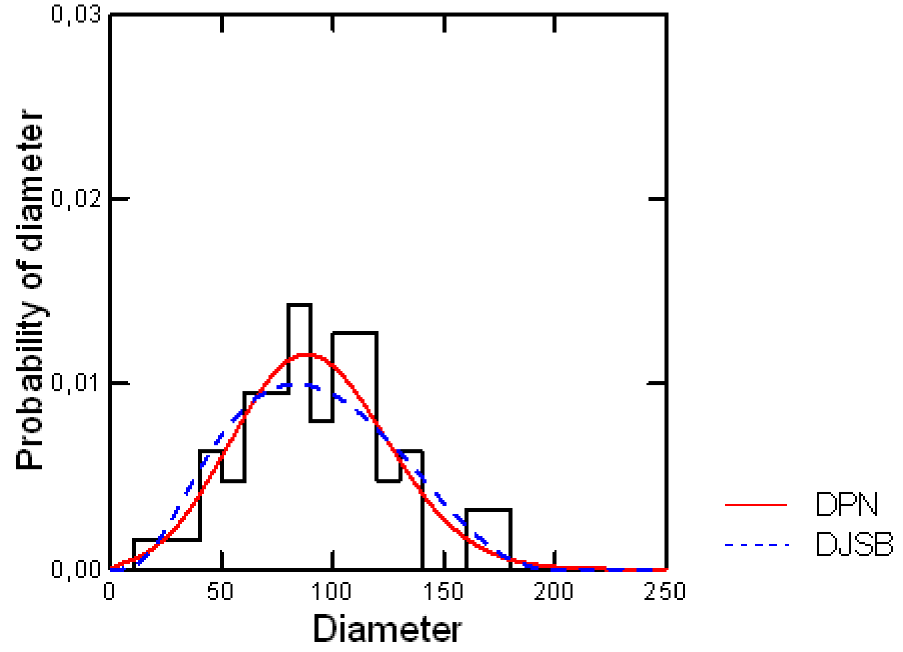
Results (↓) for STAND = 71

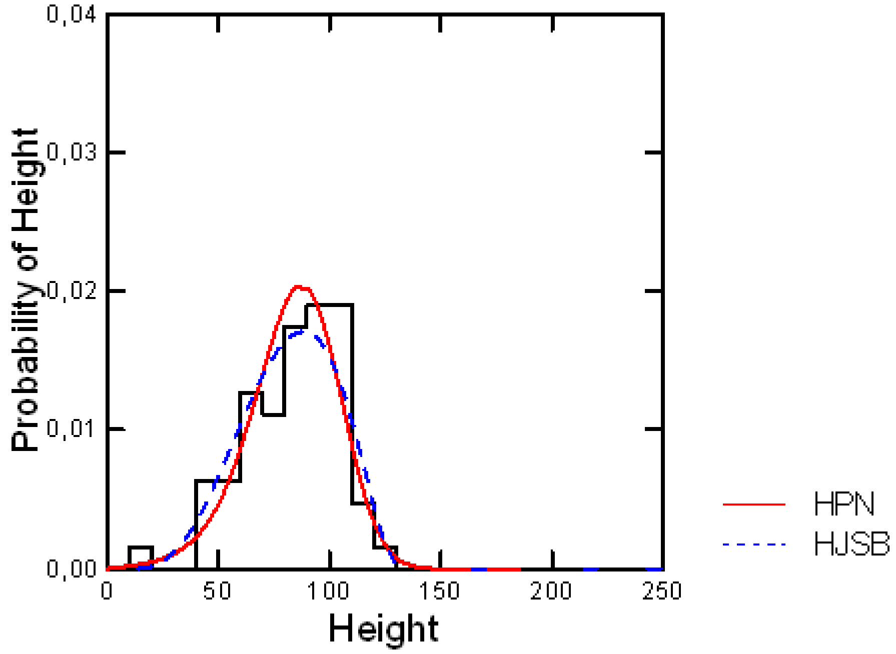


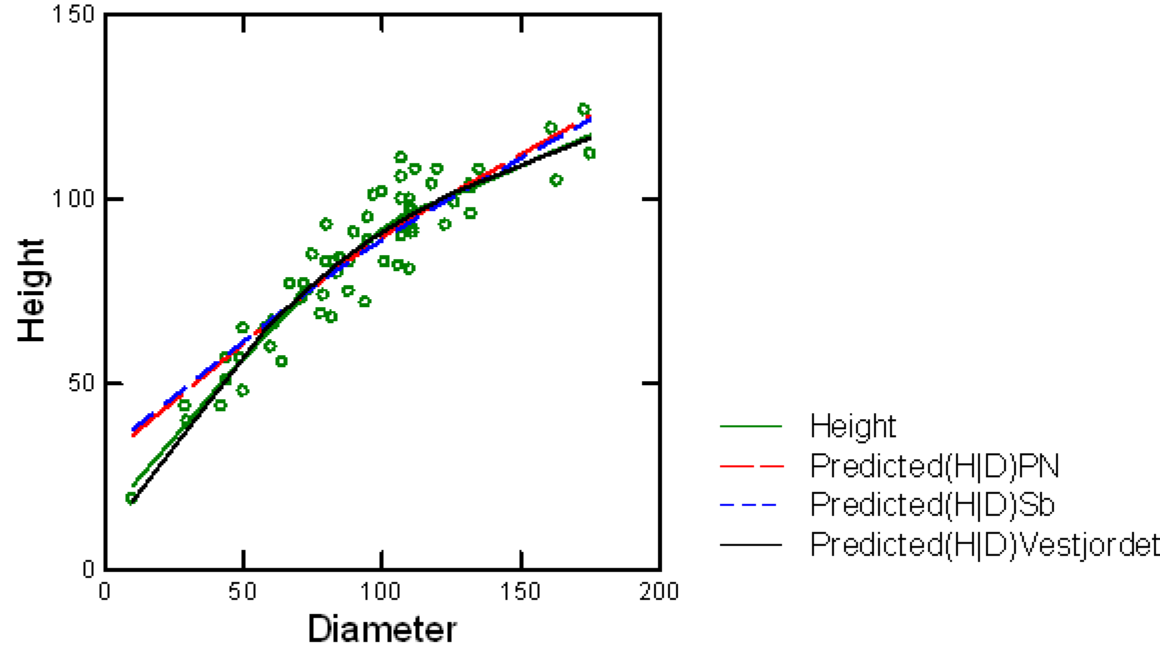




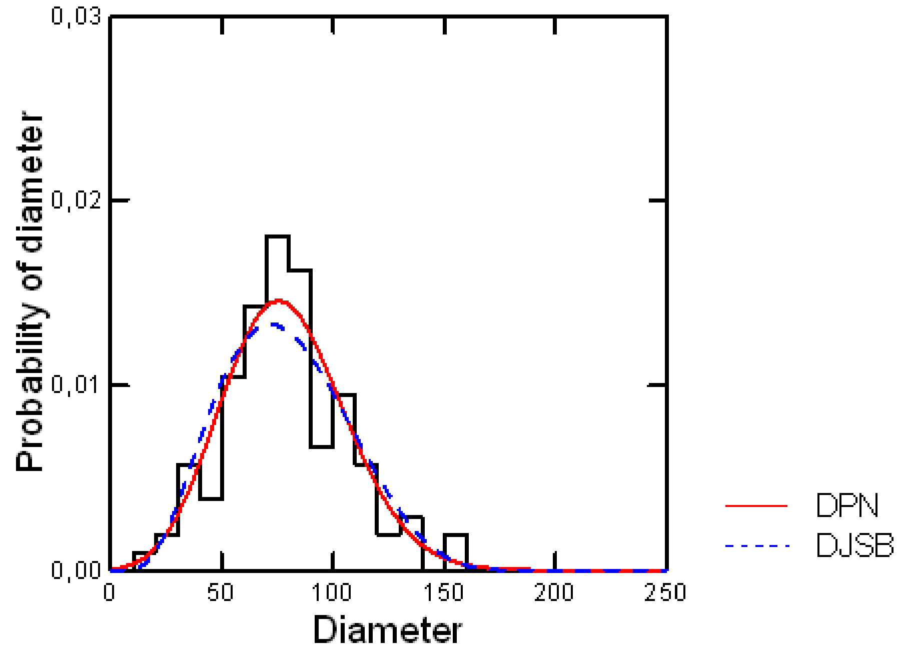
Results (↓) for STAND = 72

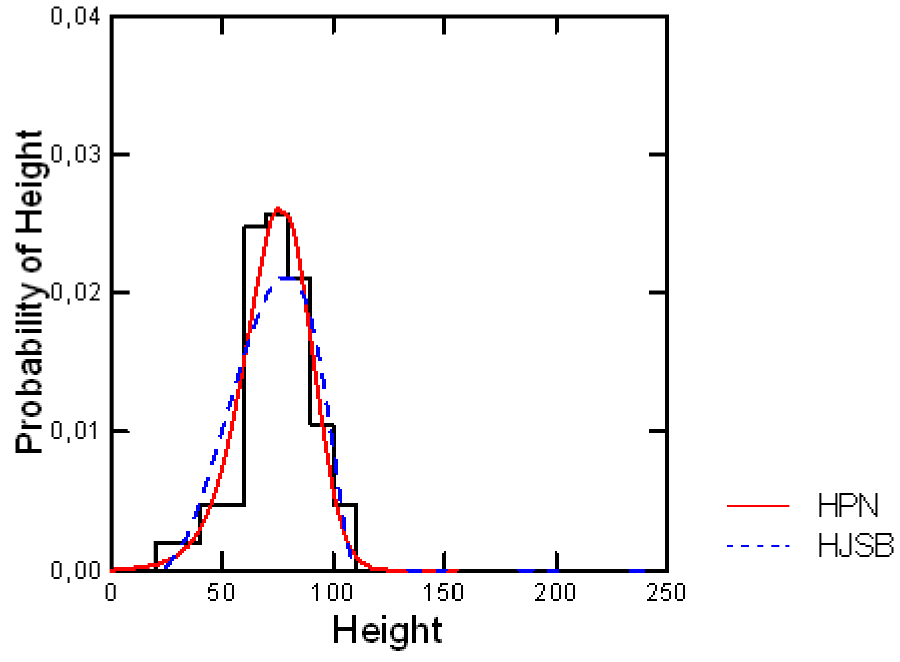


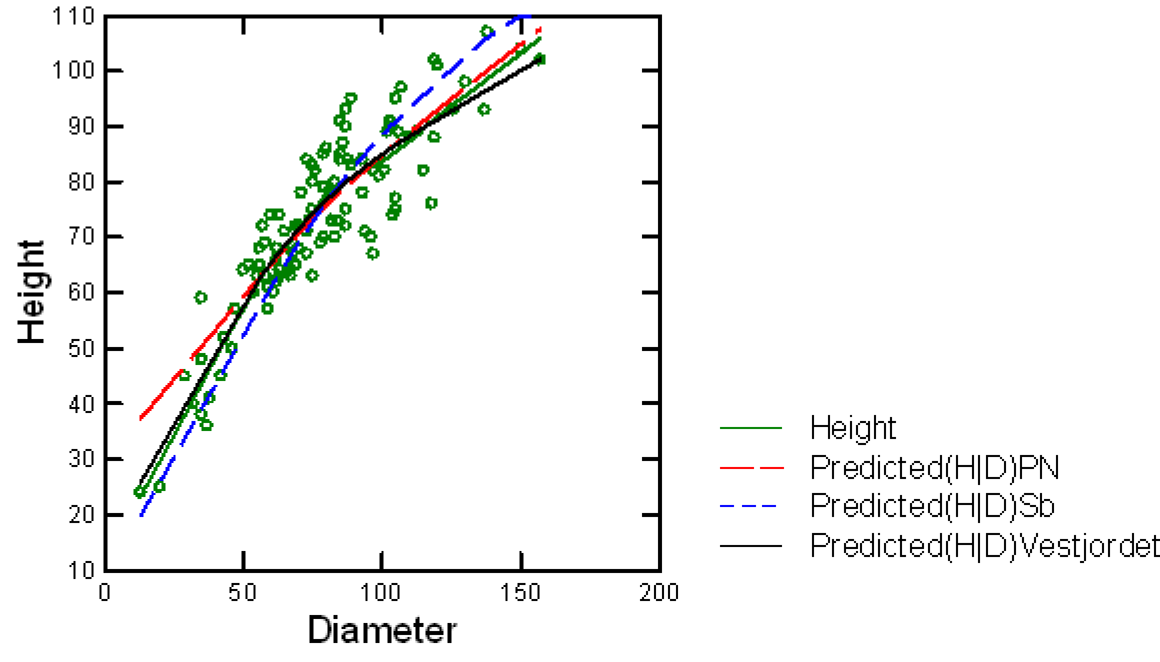




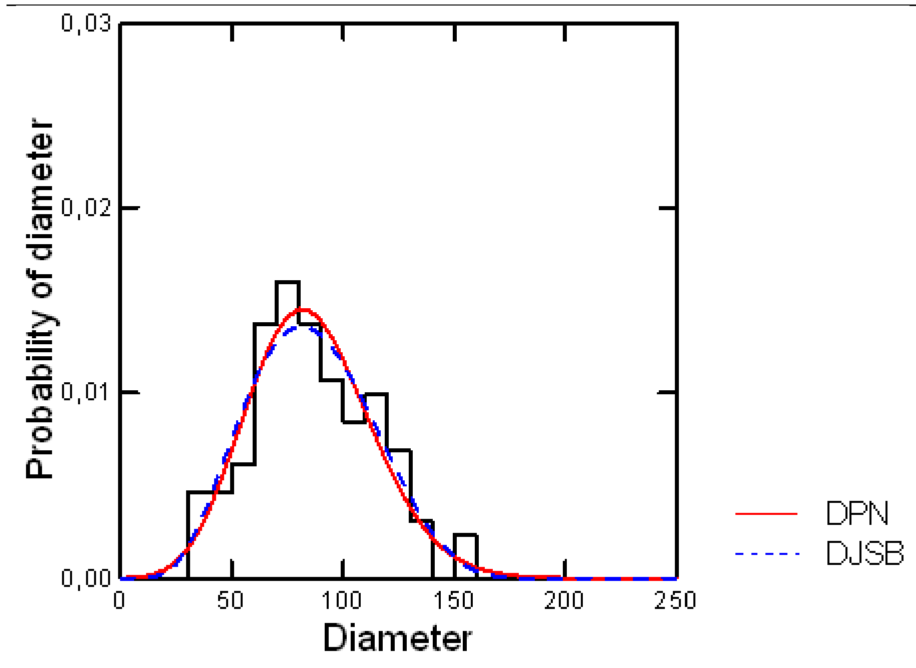
Results (↓) for STAND = 73

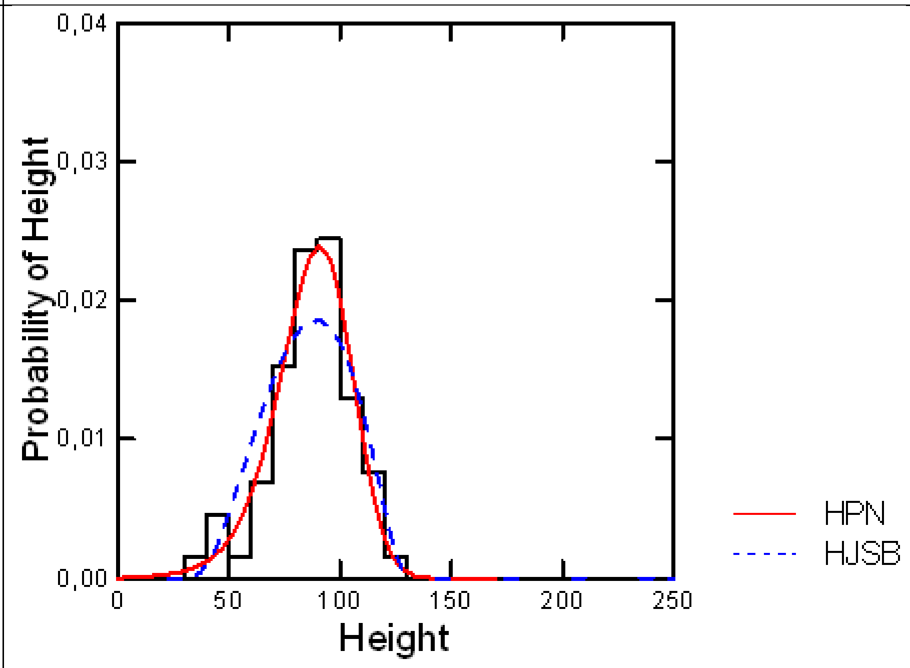


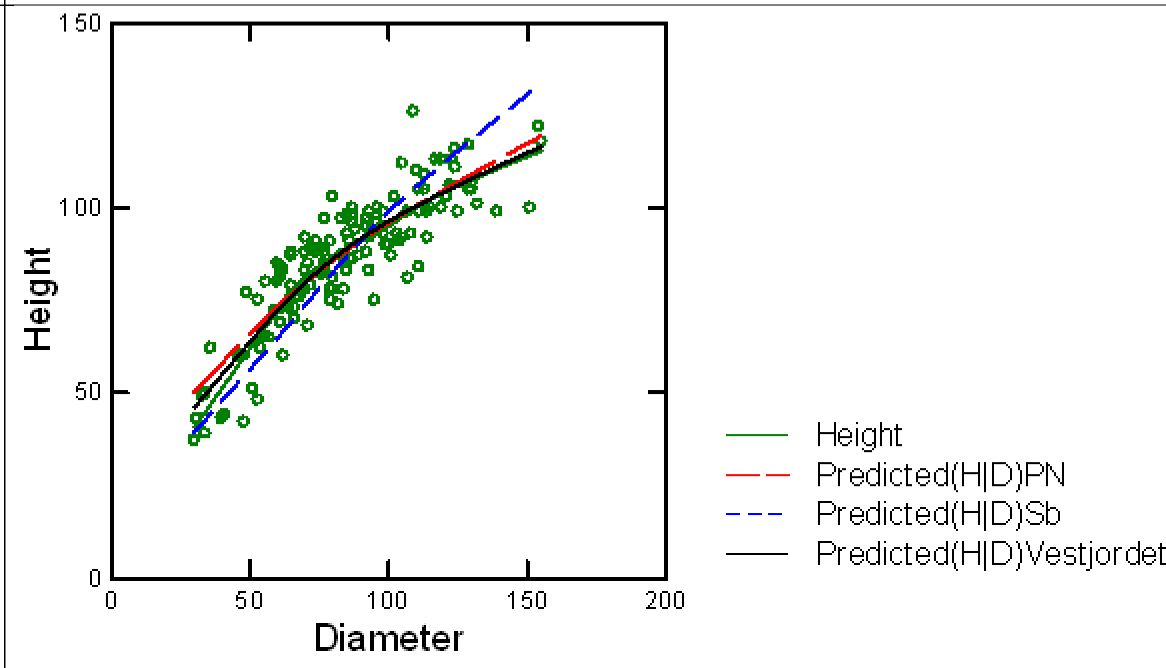




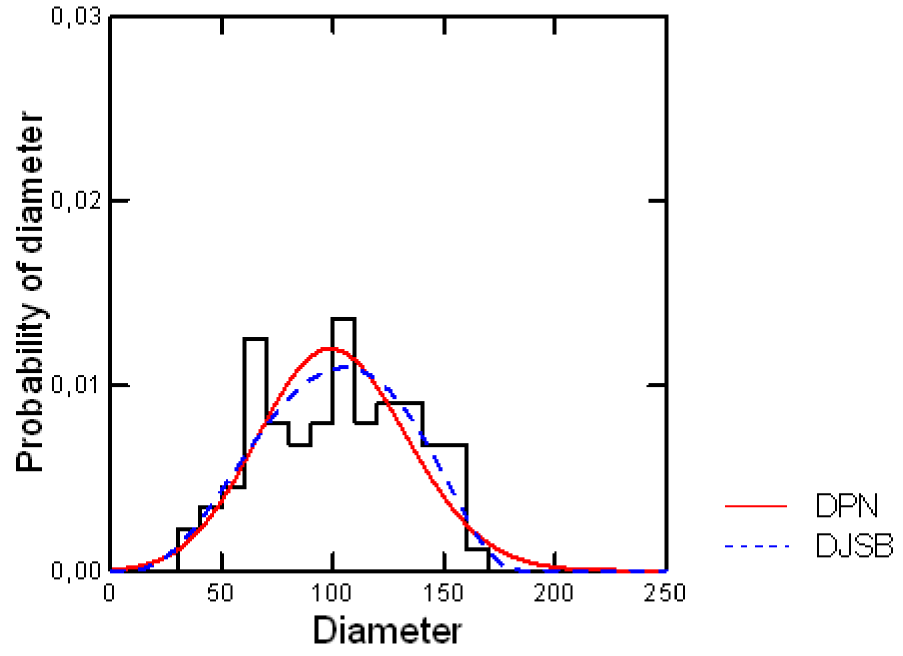
Results (↓) for STAND = 74

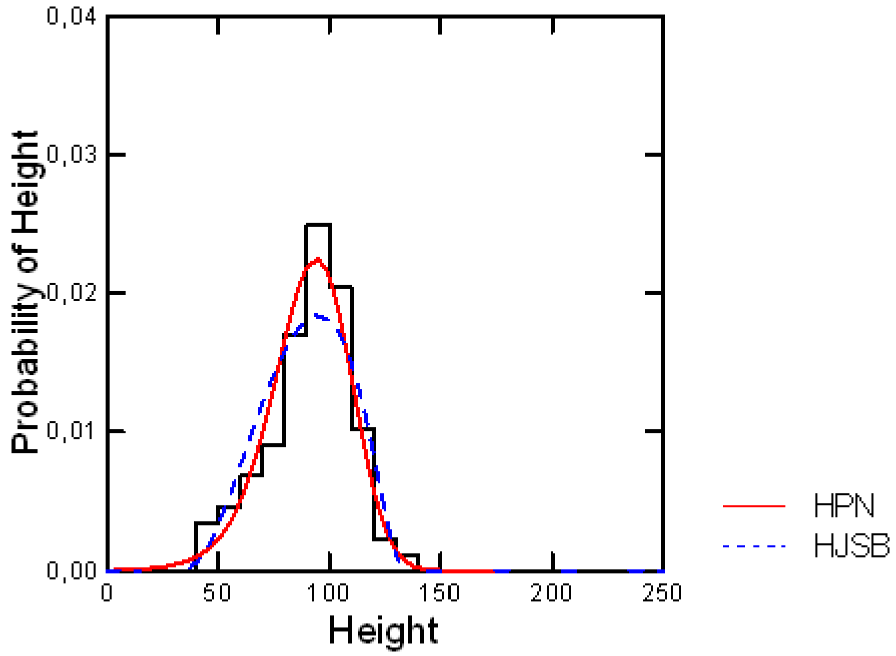


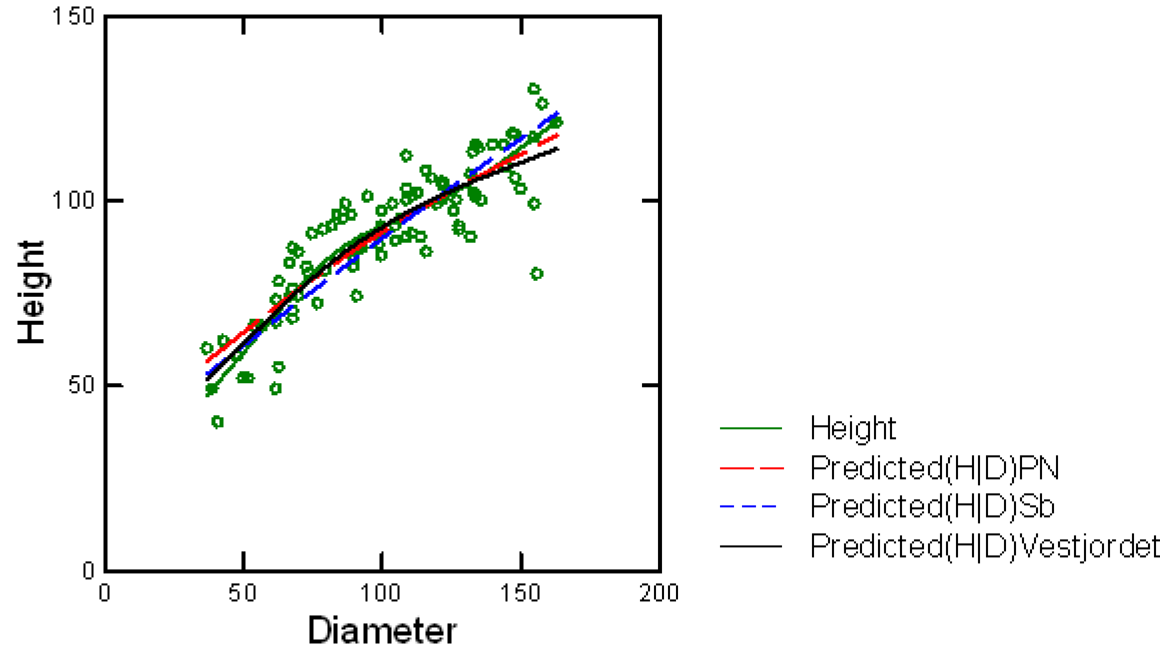




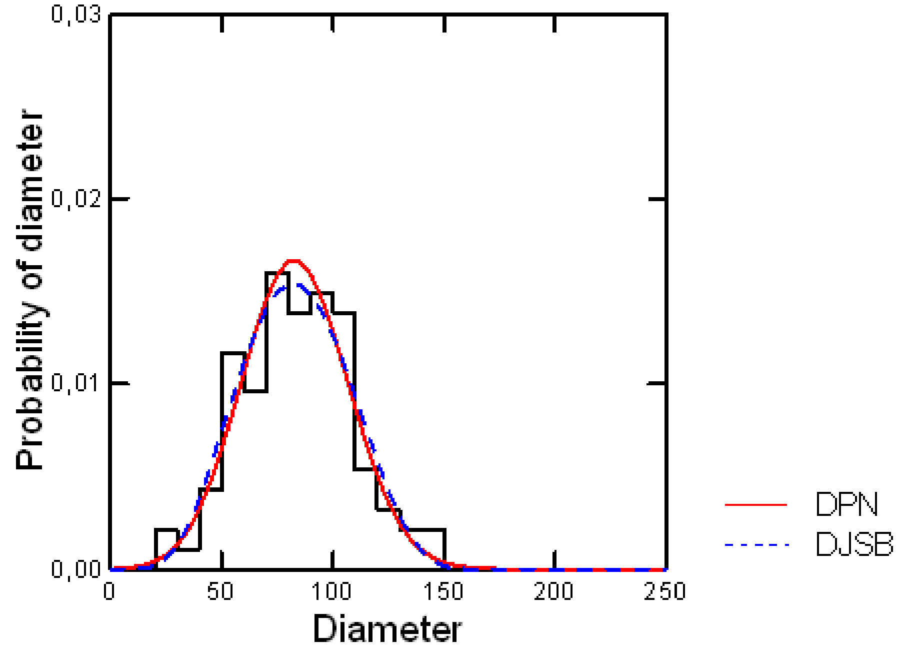
Results (↓) for STAND = 75

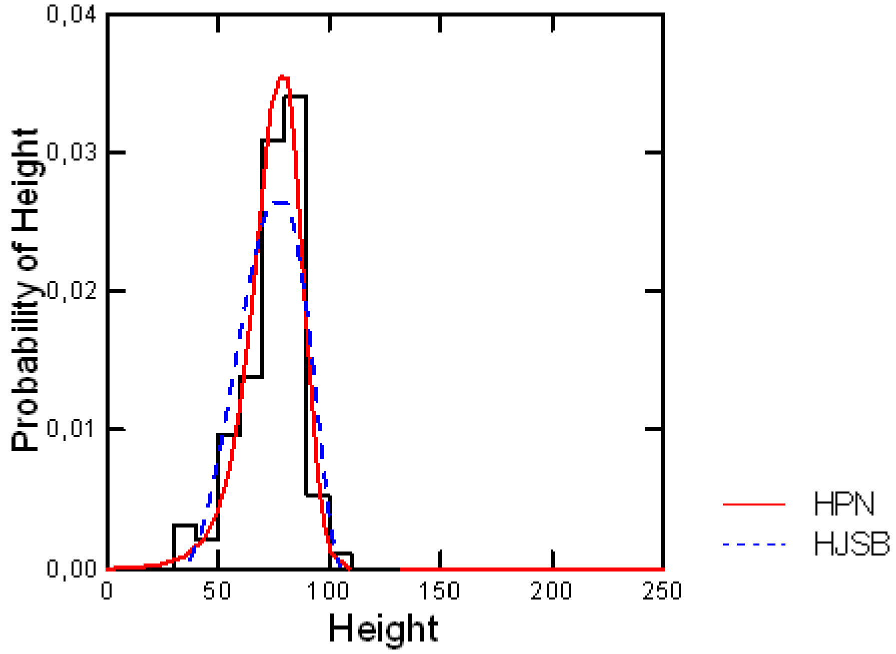


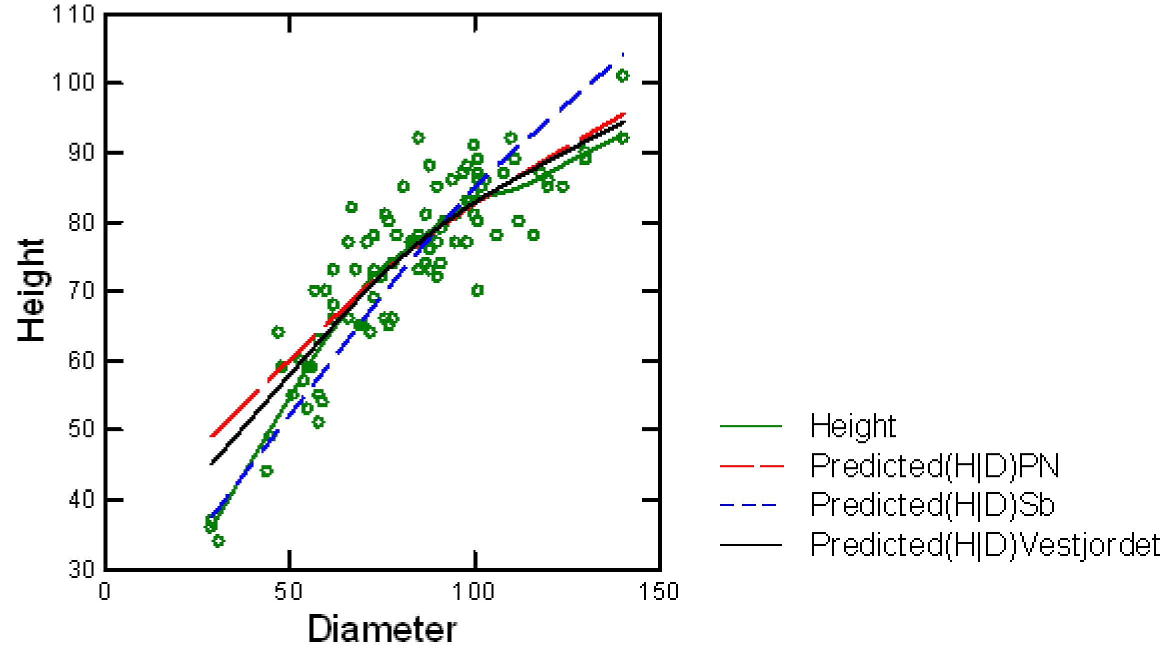




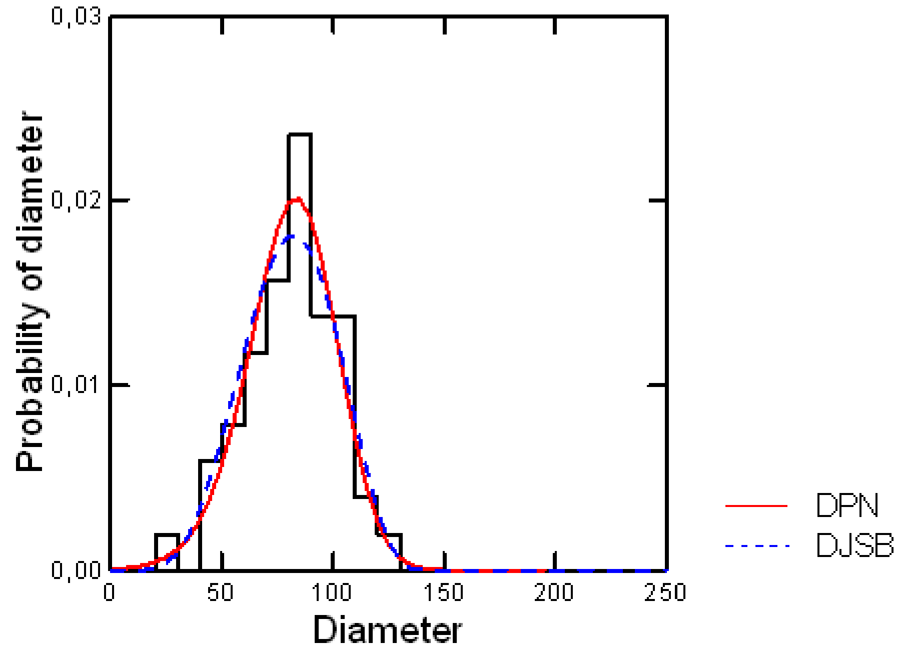
Results (↓) for STAND = 76

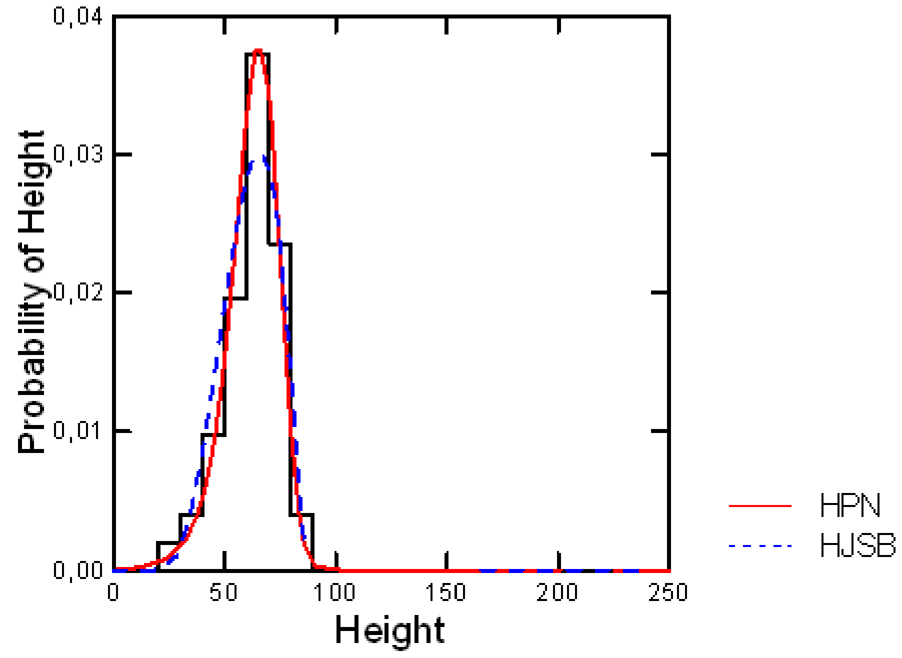


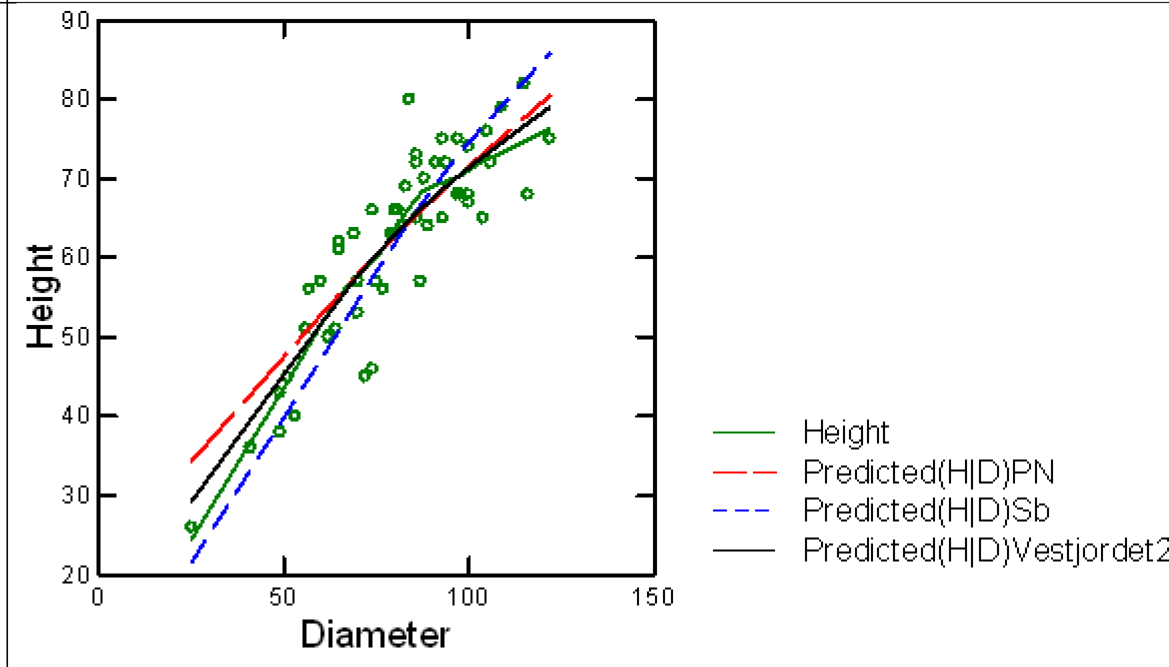




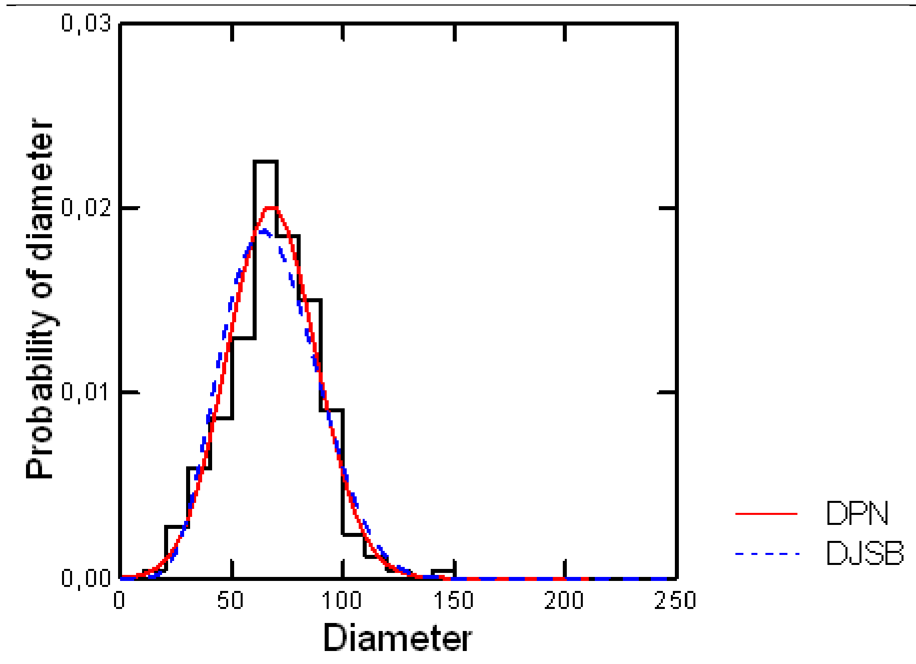
Results (↓) for STAND = 77

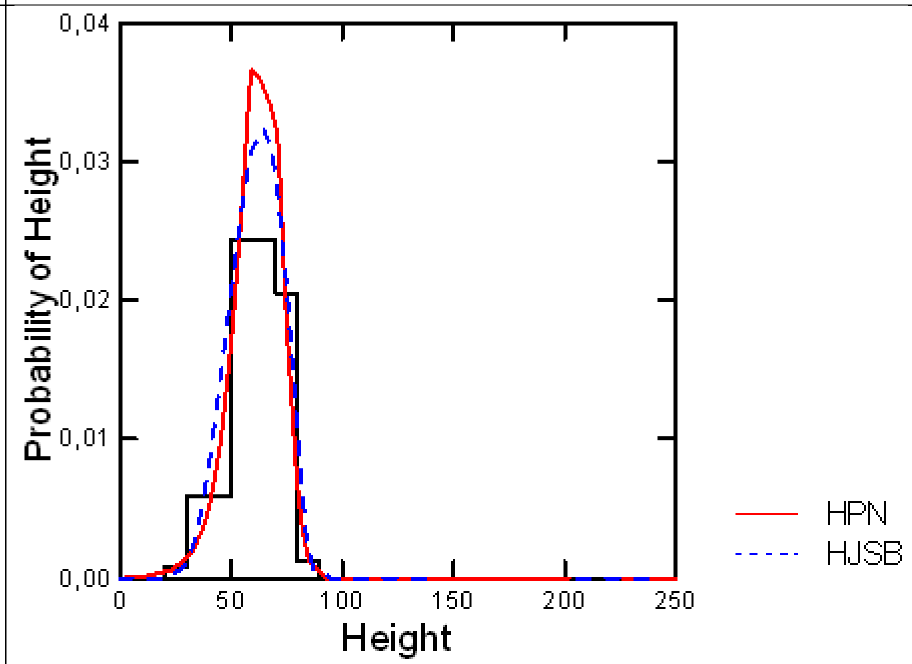


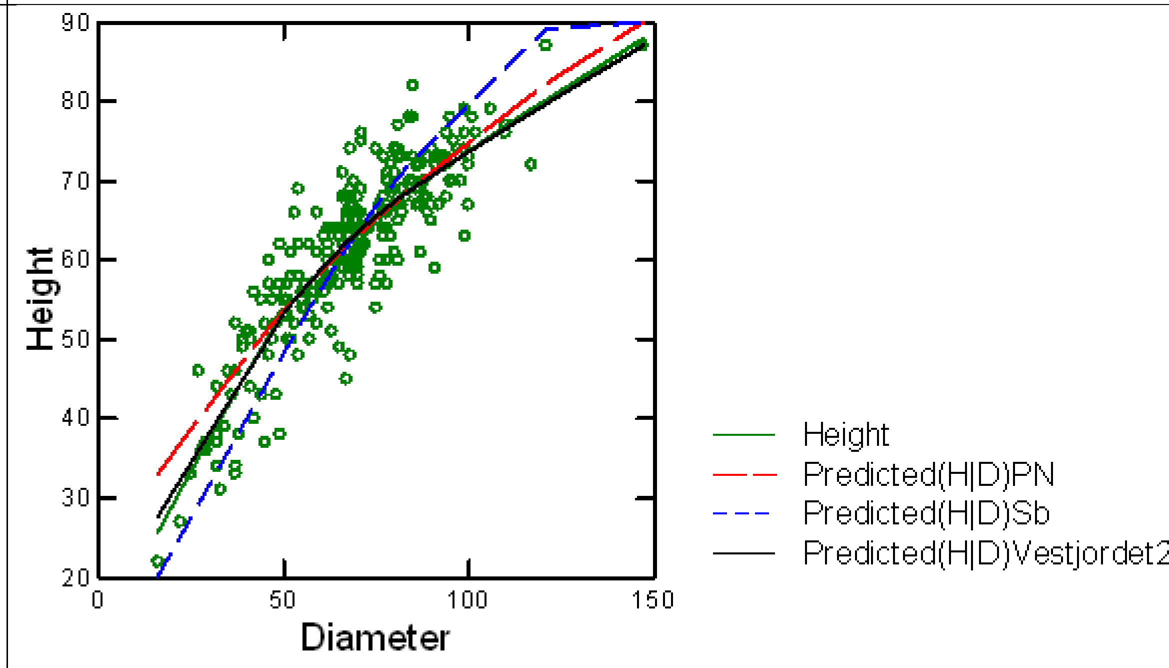




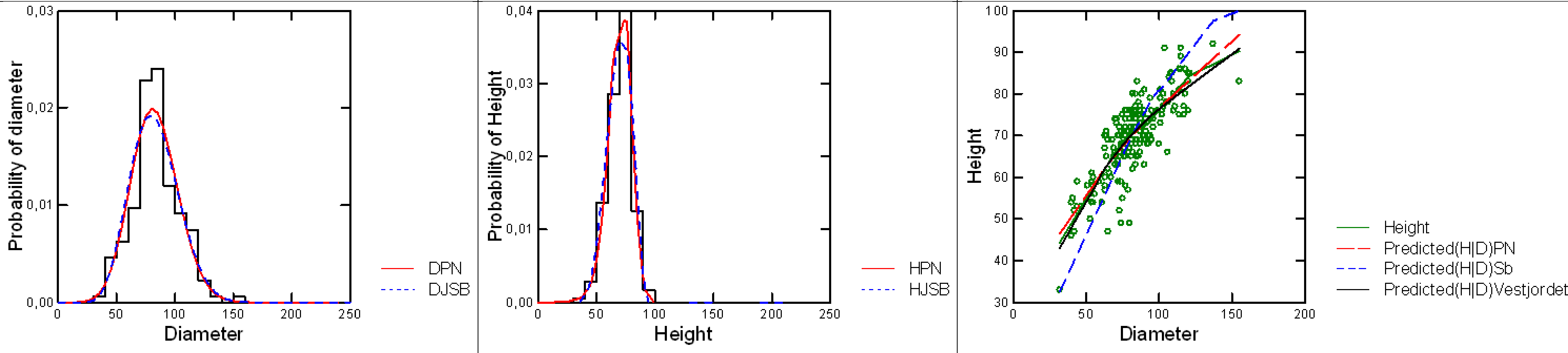
Results (↓) for STAND = 78



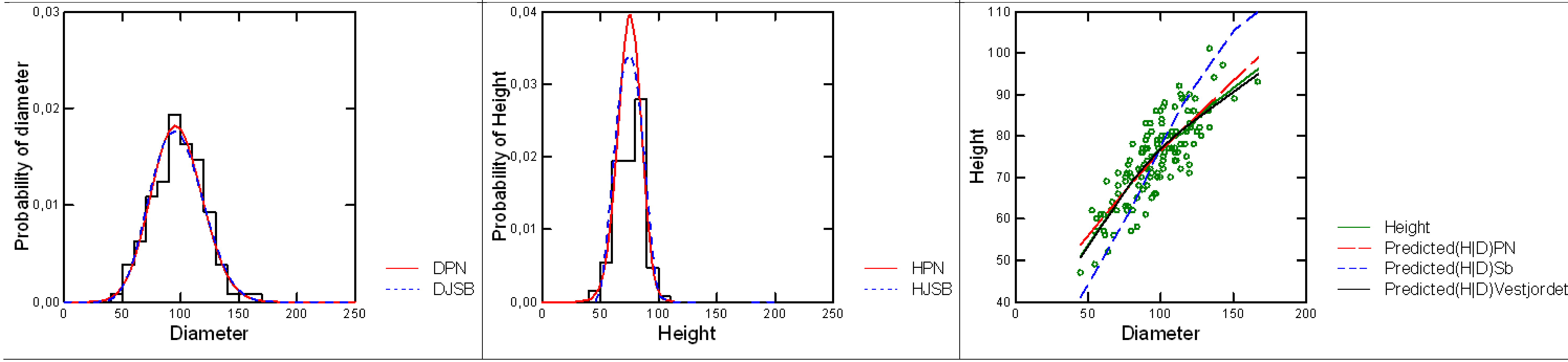




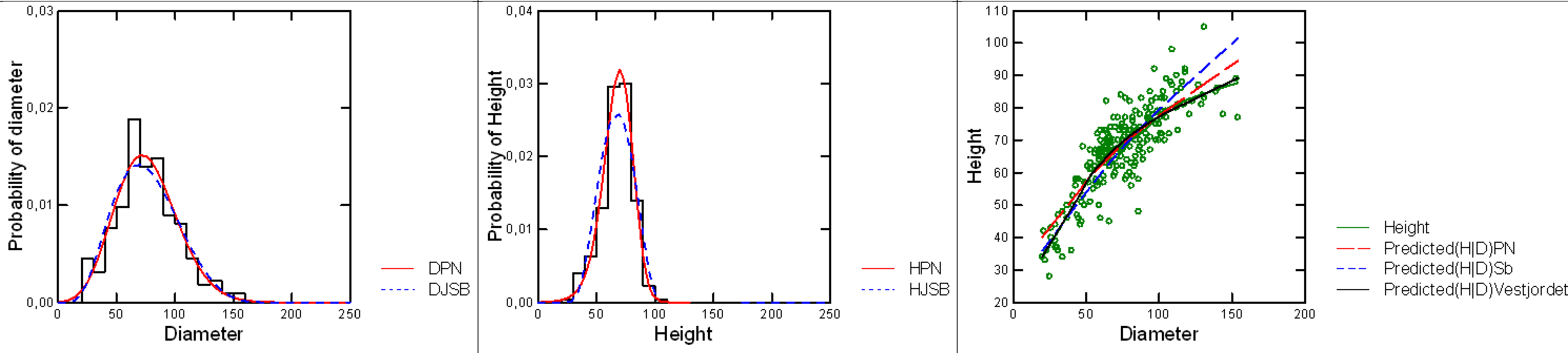
Results (↓) for STAND = 79



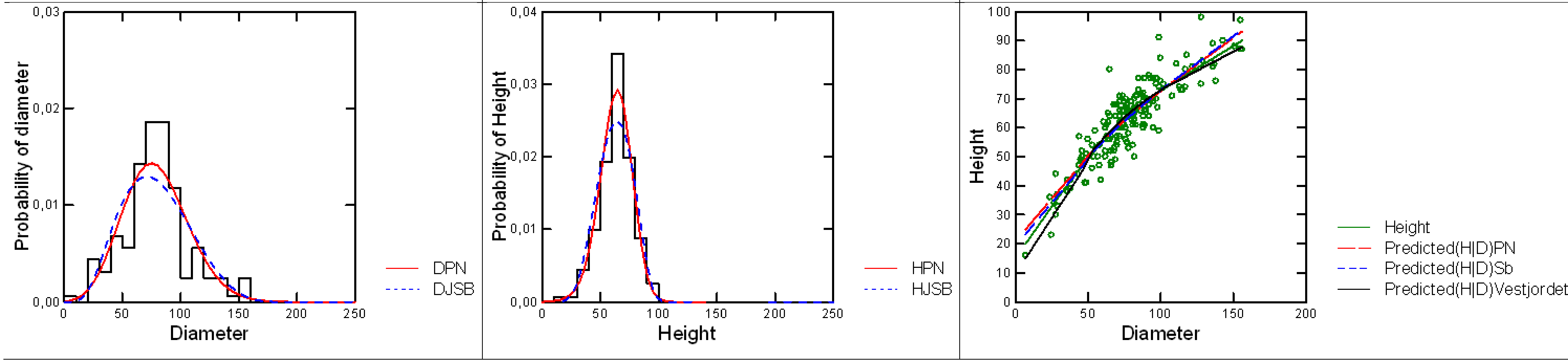
Results (↓) for STAND = 80



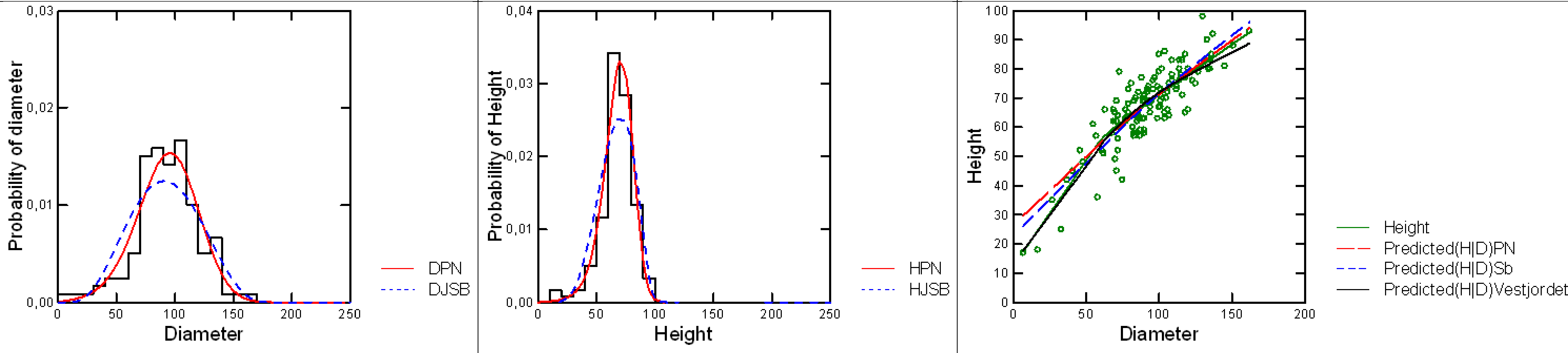
Results (↓) for STAND = 81



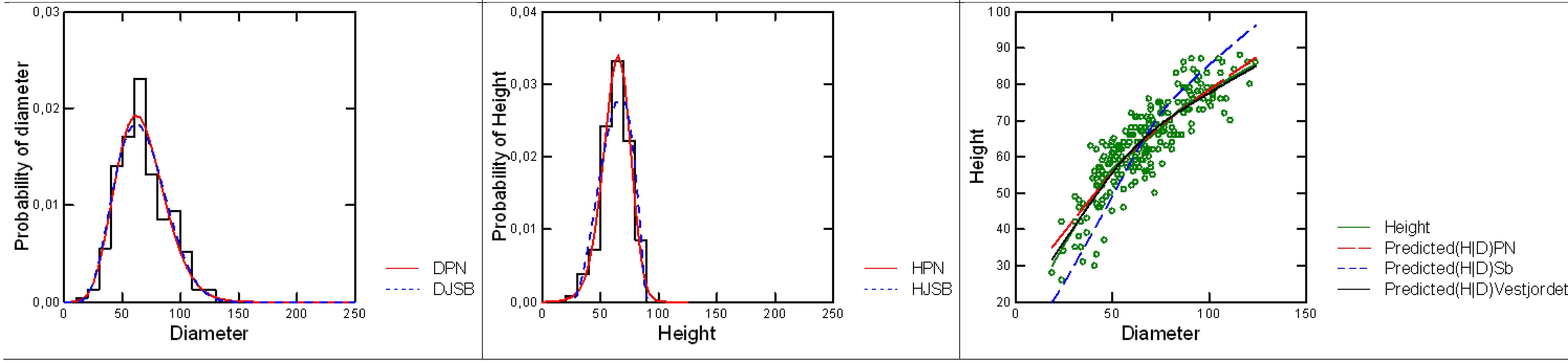
Results (↓) for STAND = 82



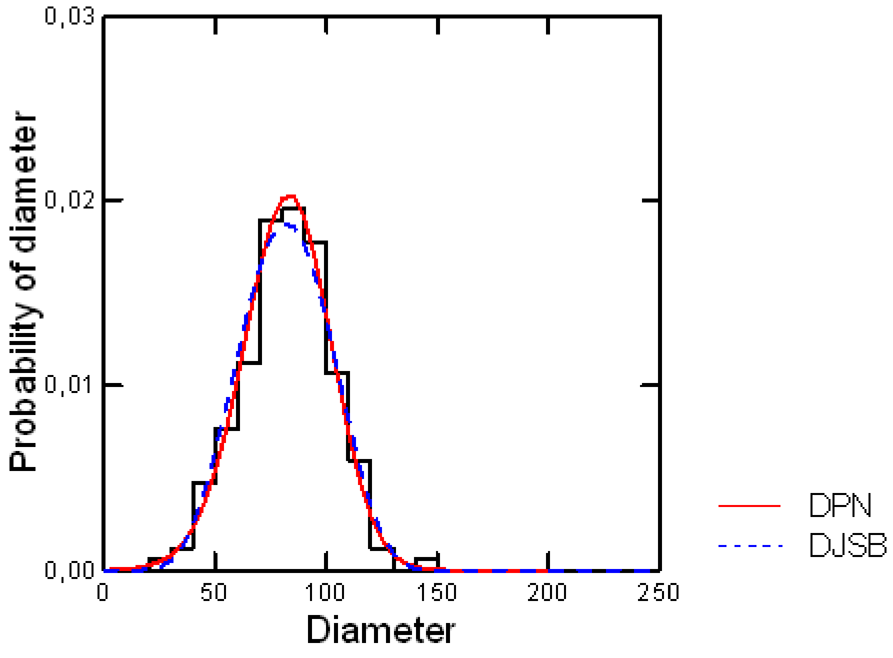
Results (↓) for STAND = 83

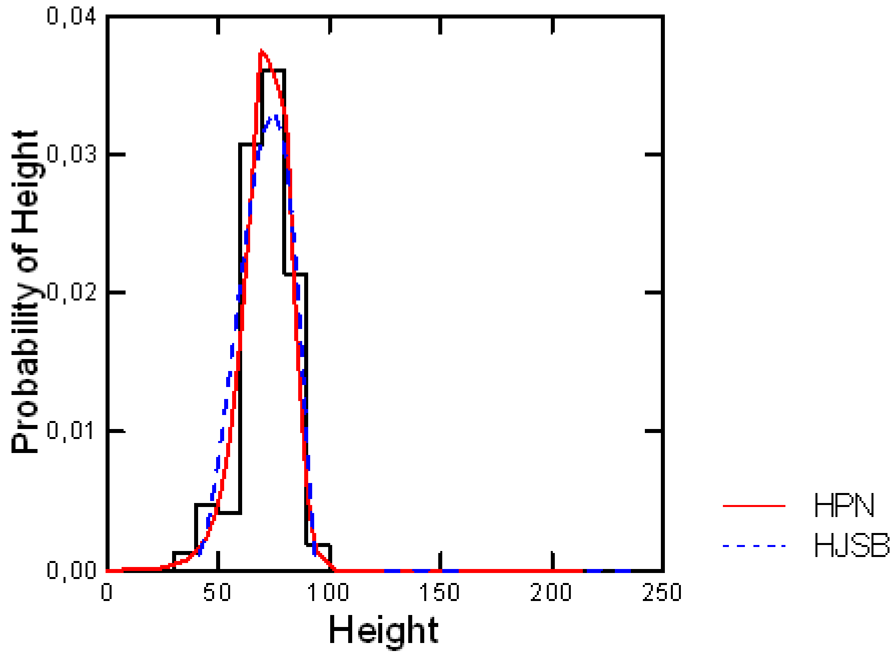


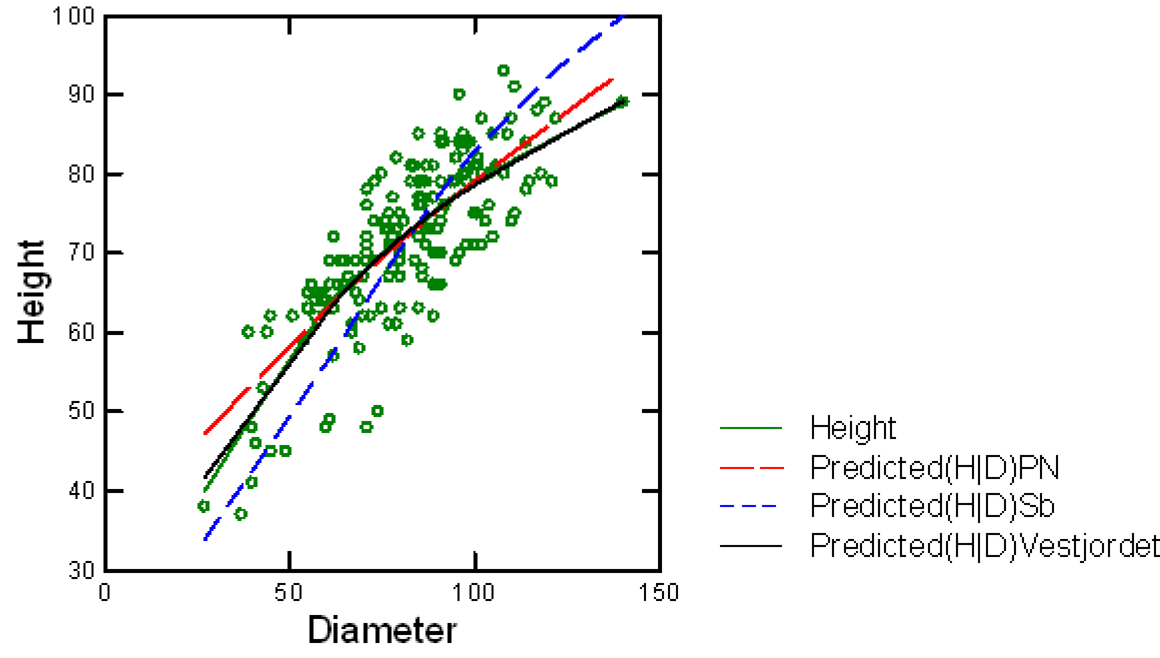
Results (↓) for STAND = 84



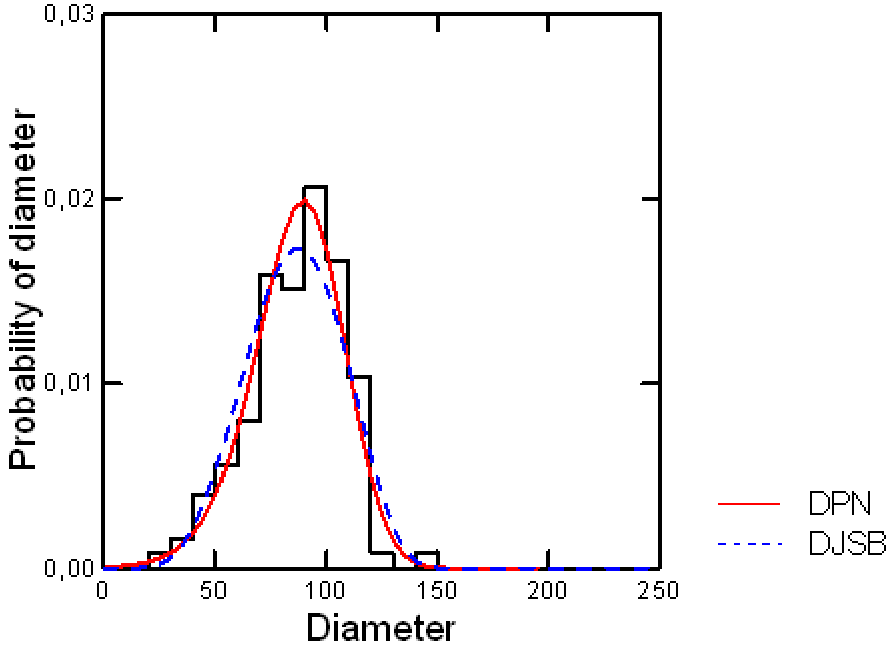
Results (↓) for STAND = 85

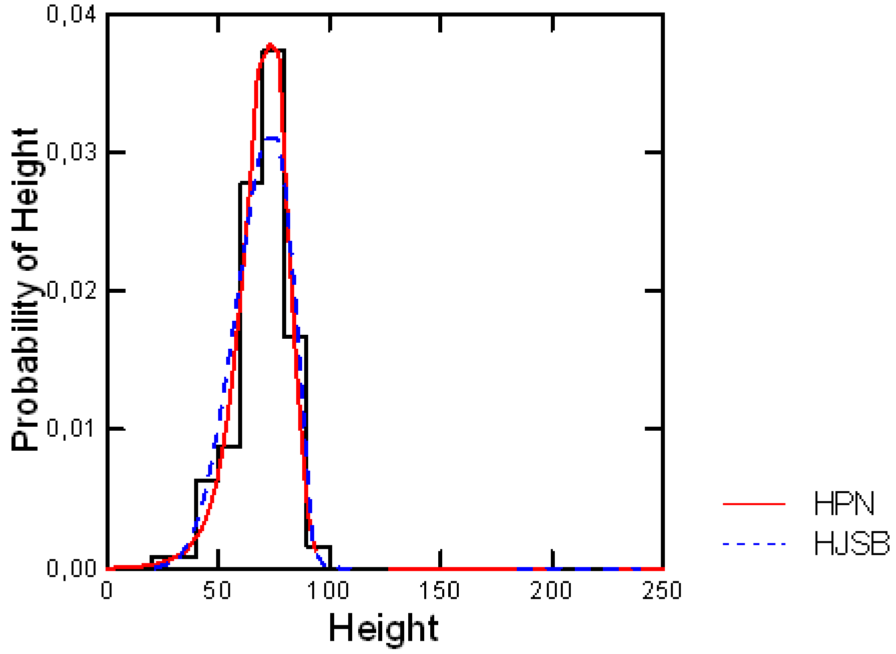


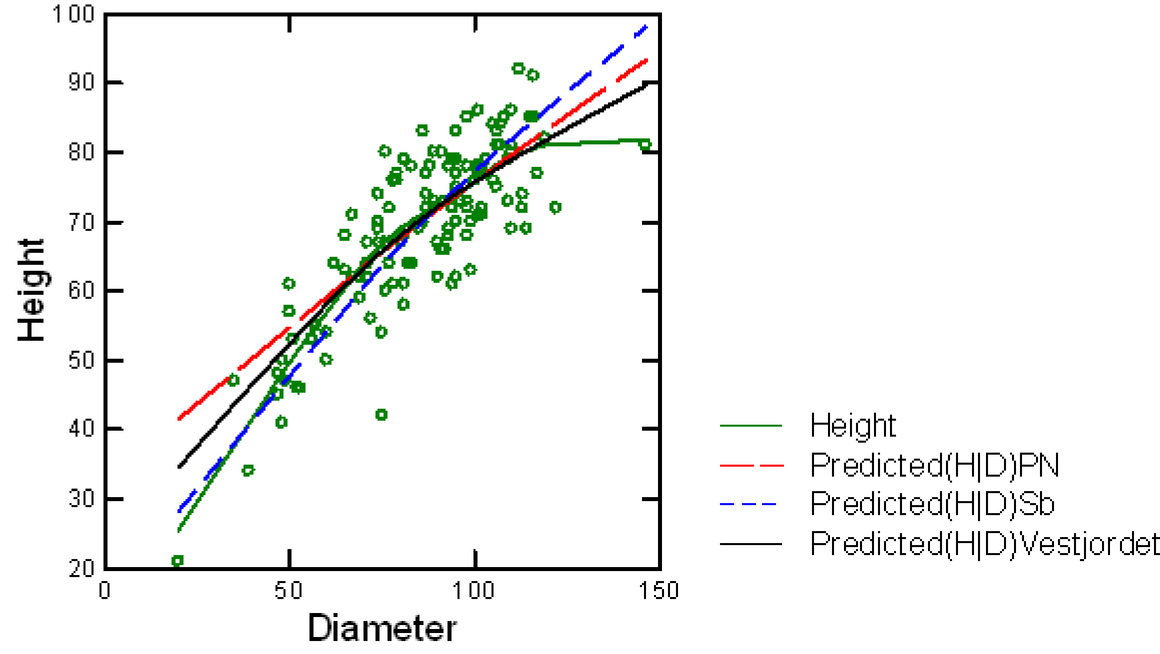




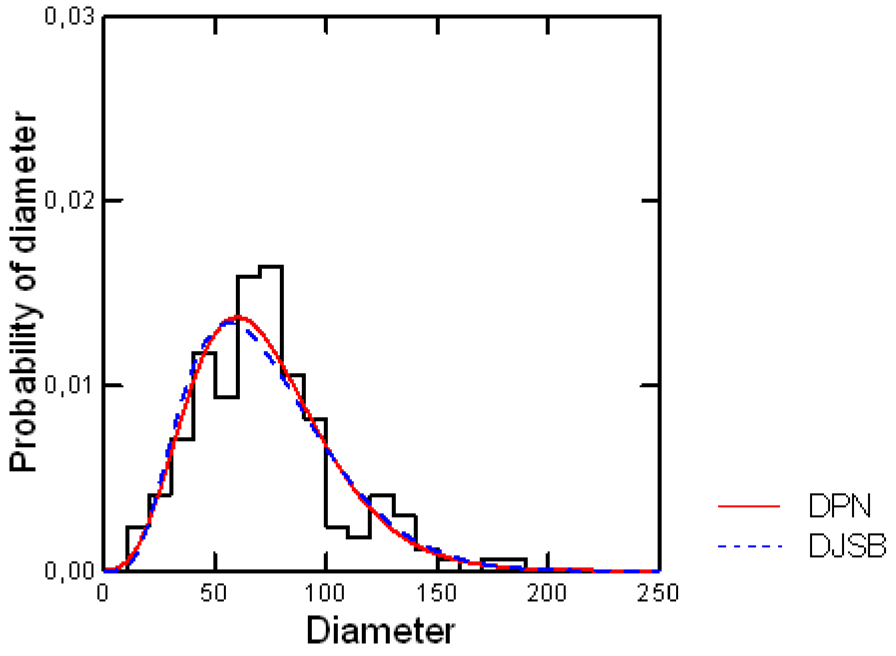
Results (↓) for STAND = 86

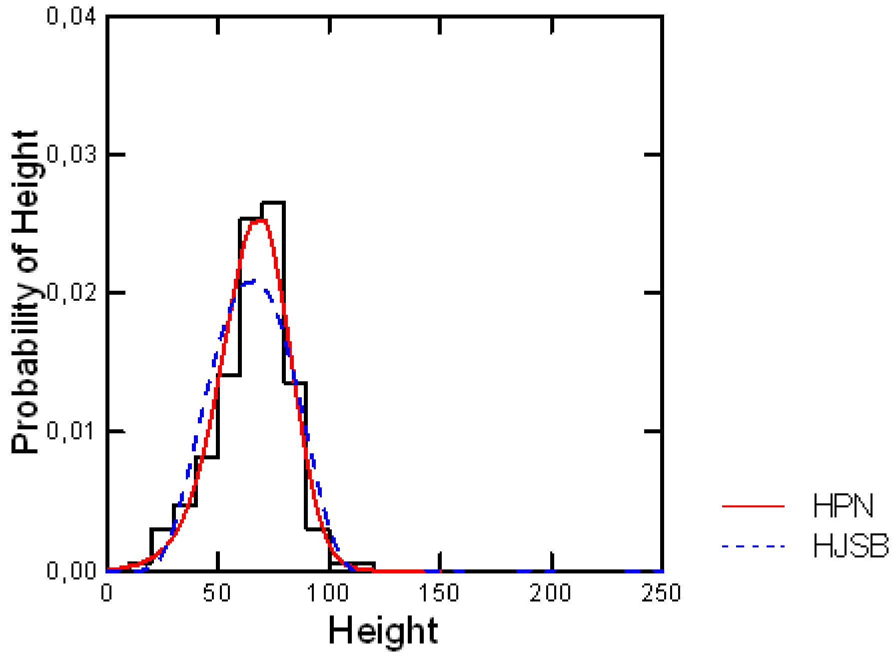


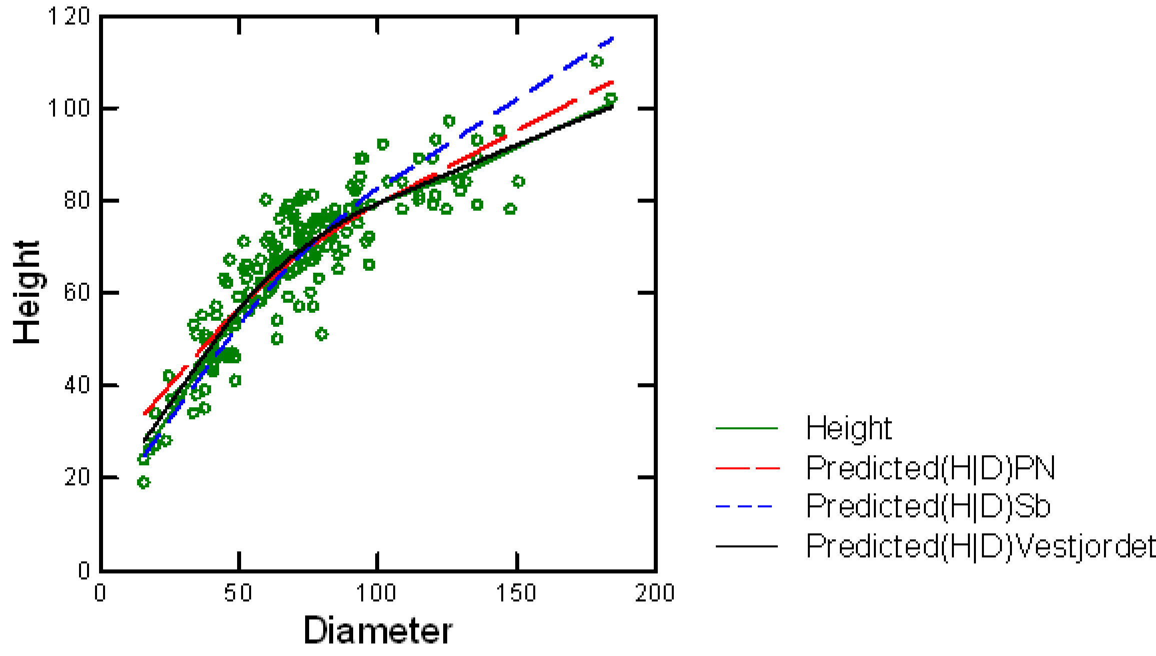




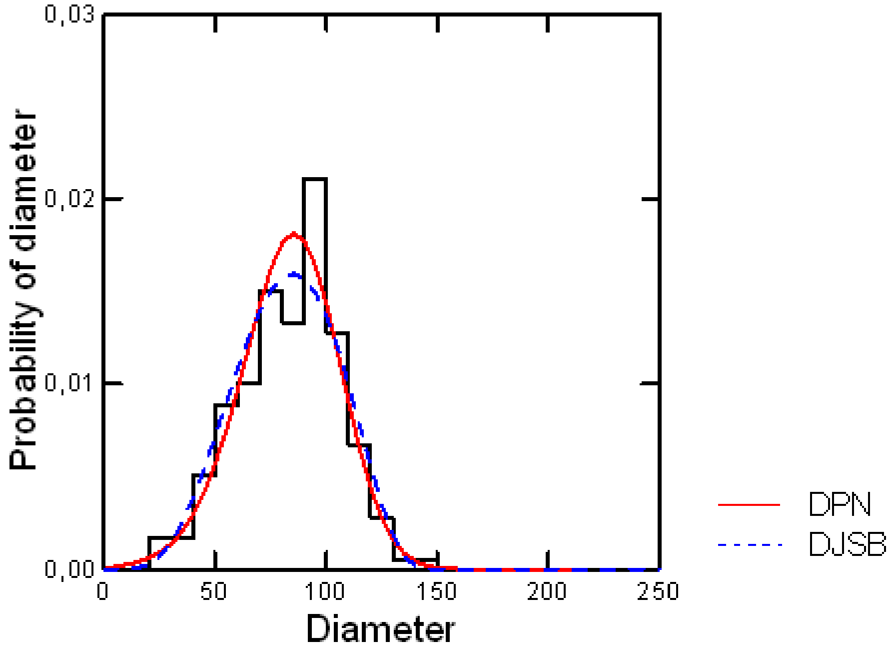
Results (↓) for STAND = 87

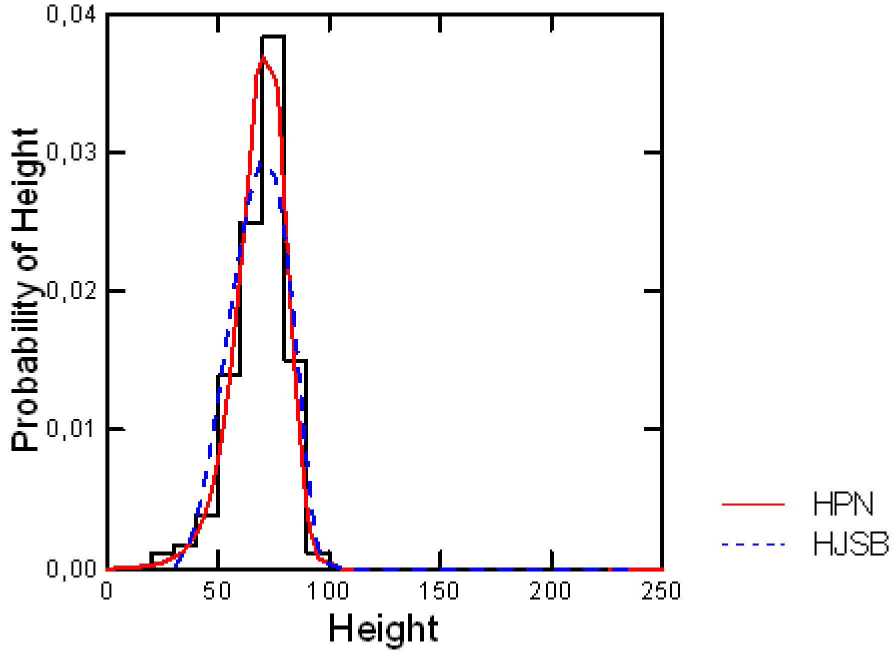


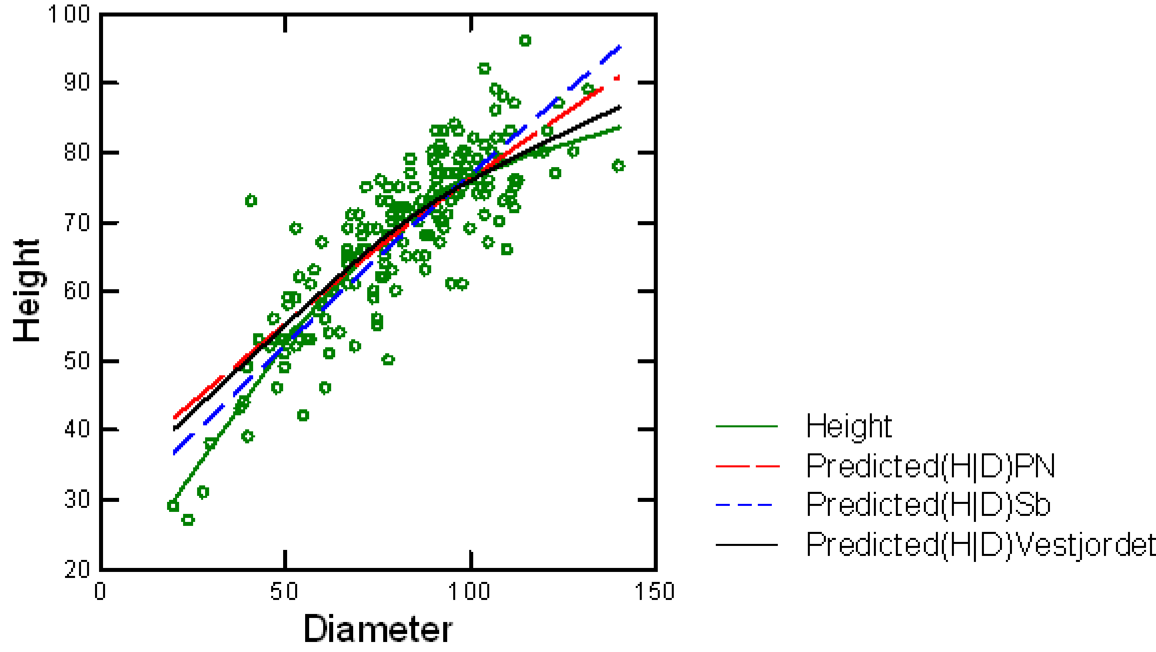




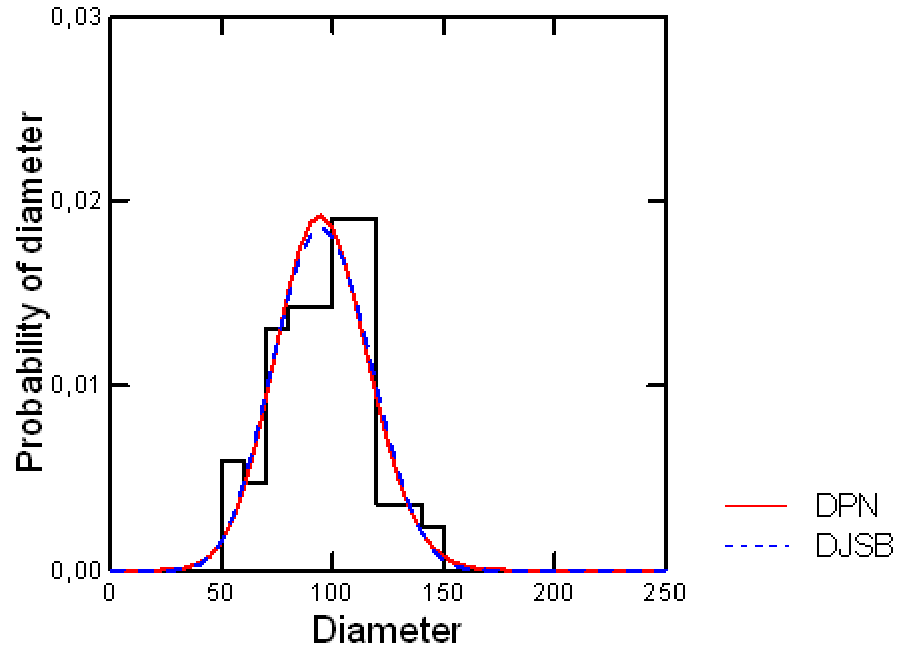
Results (↓) for STAND = 88

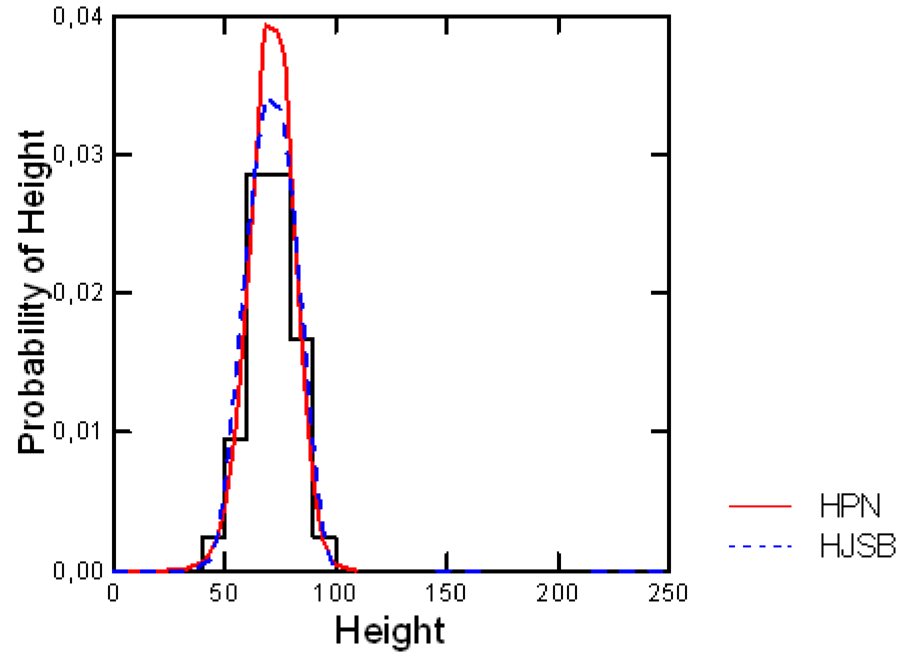


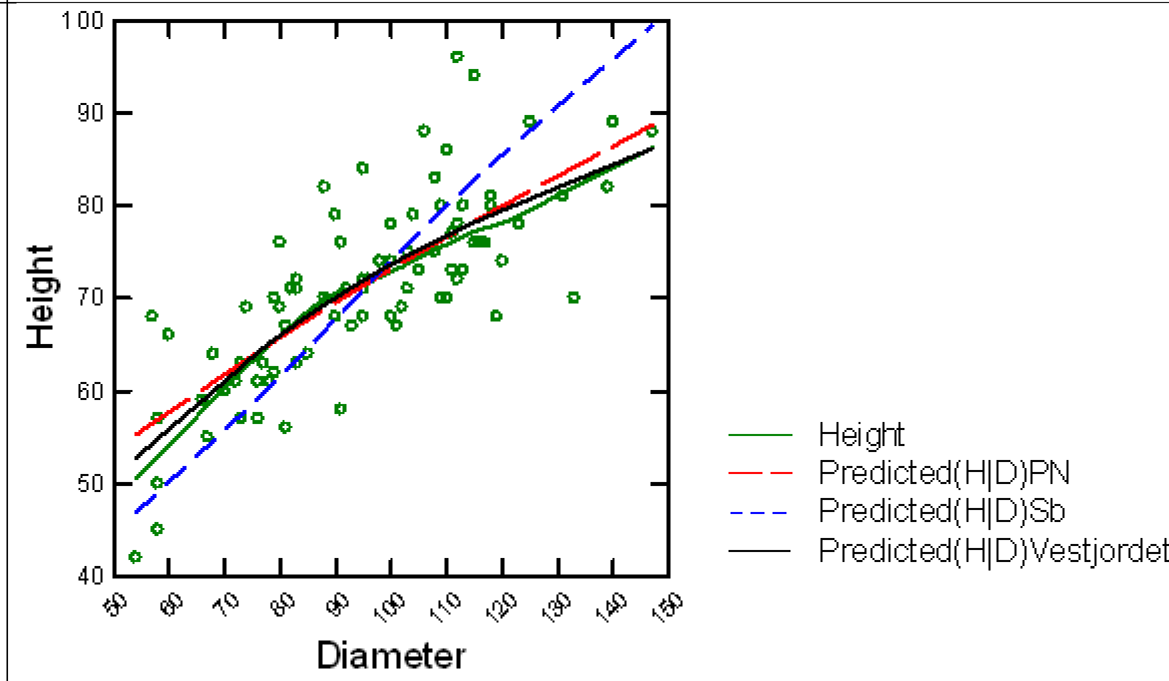




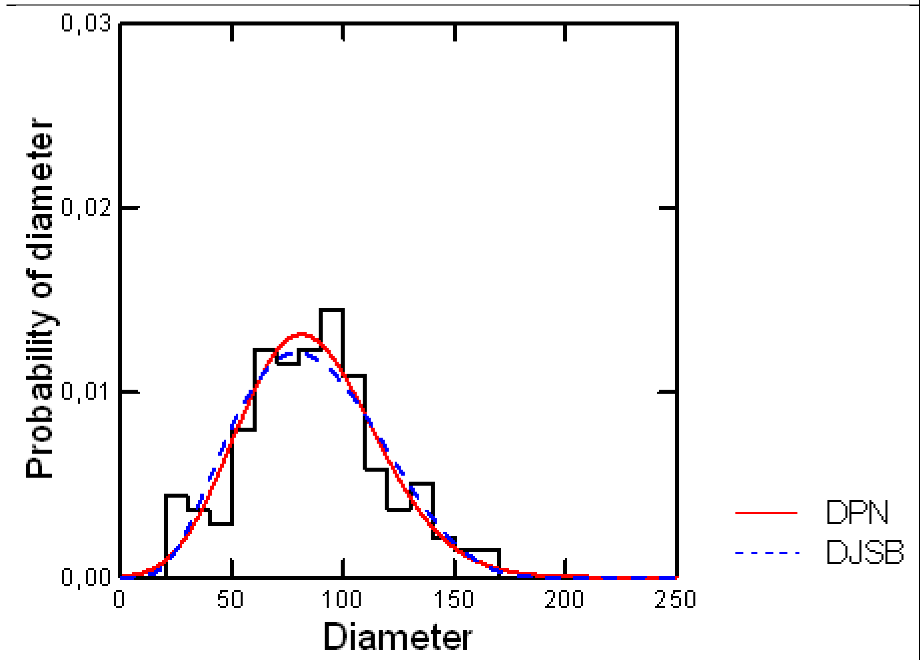
Results (↓) for STAND = 89

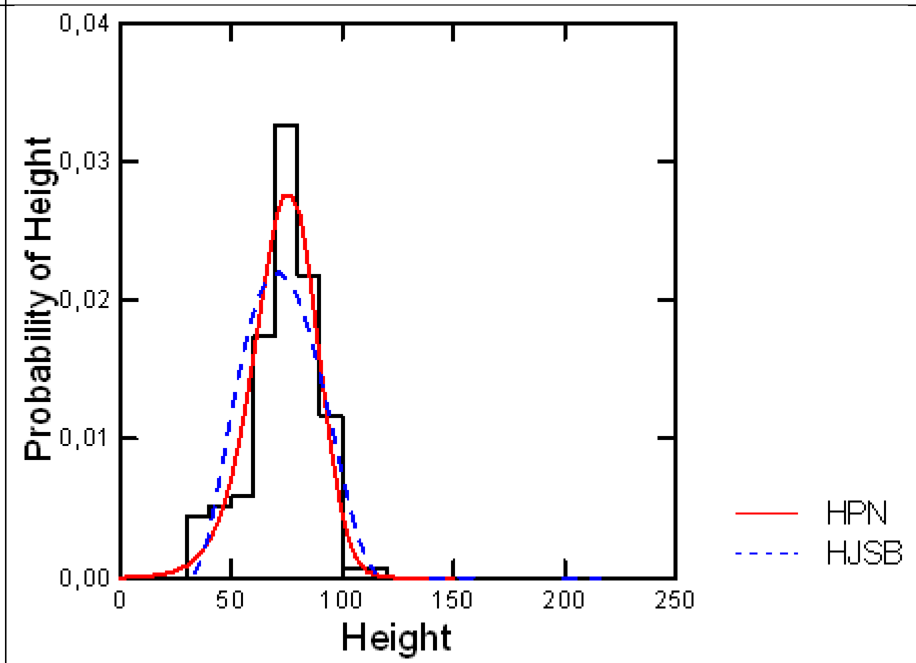


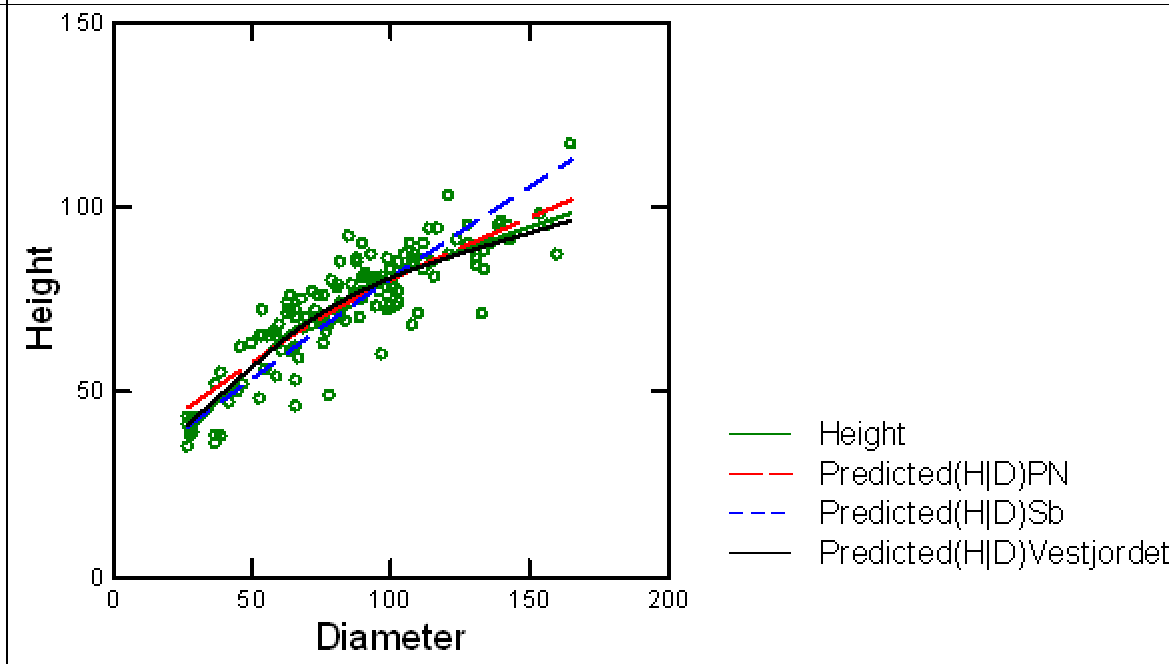




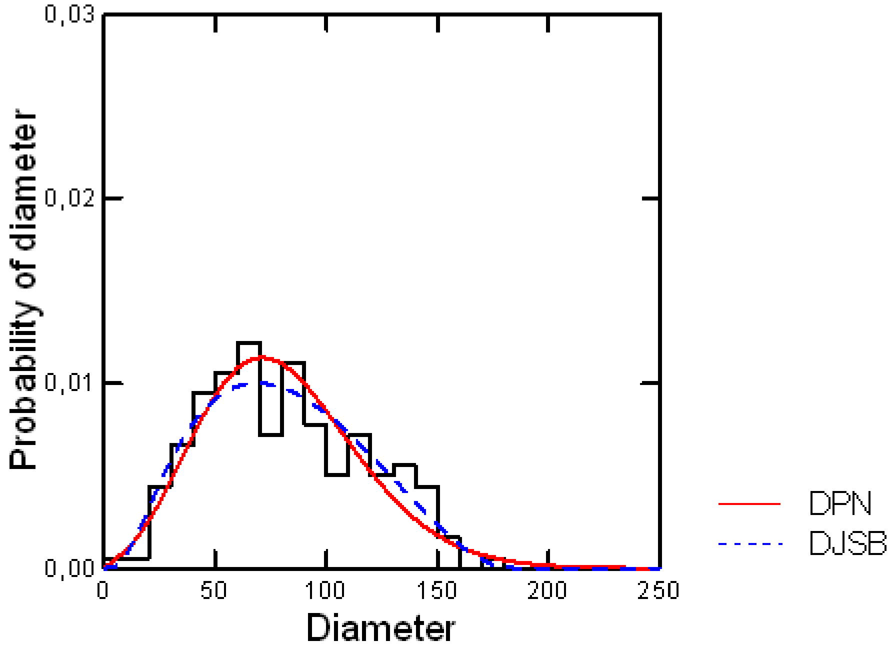
Results (↓) for STAND = 90

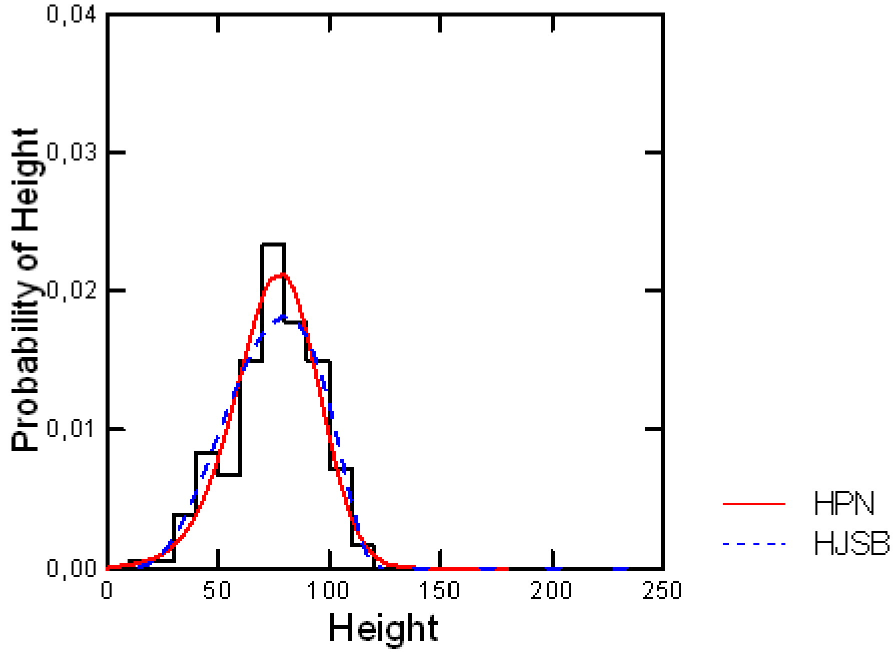


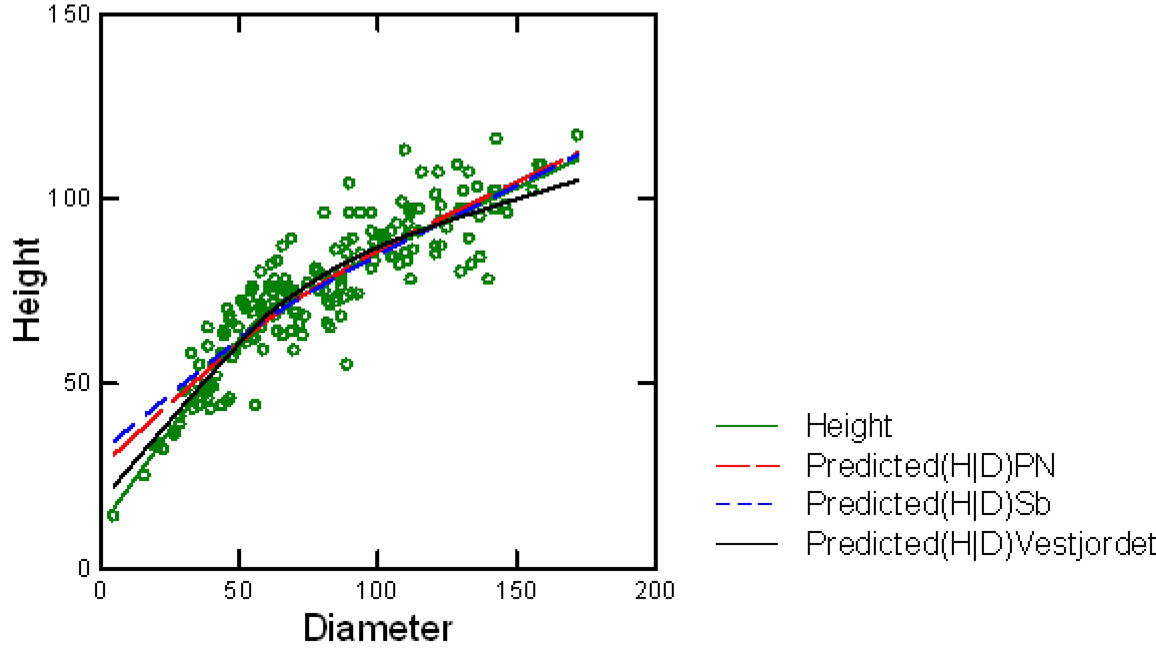




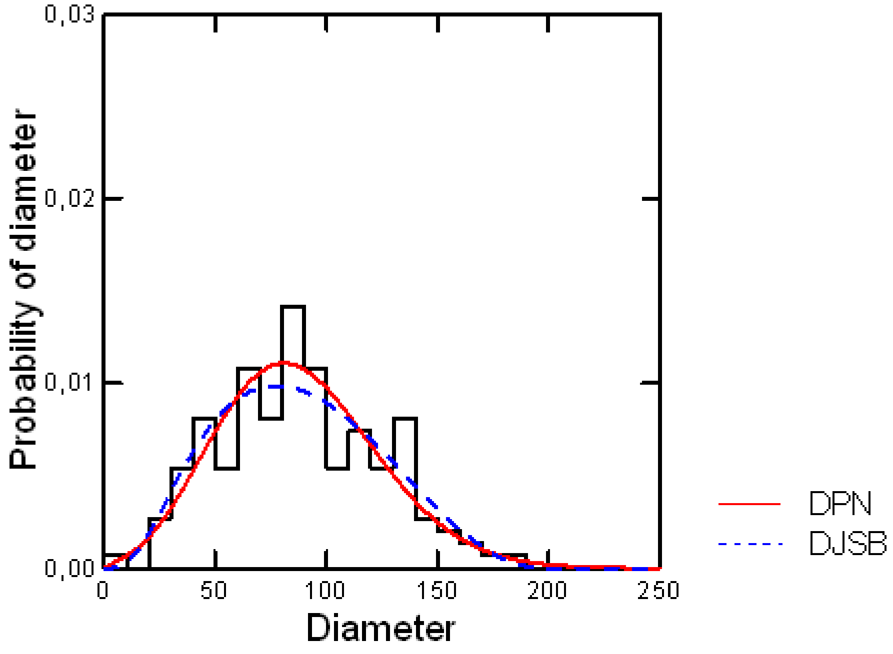
Results (↓) for STAND = 91

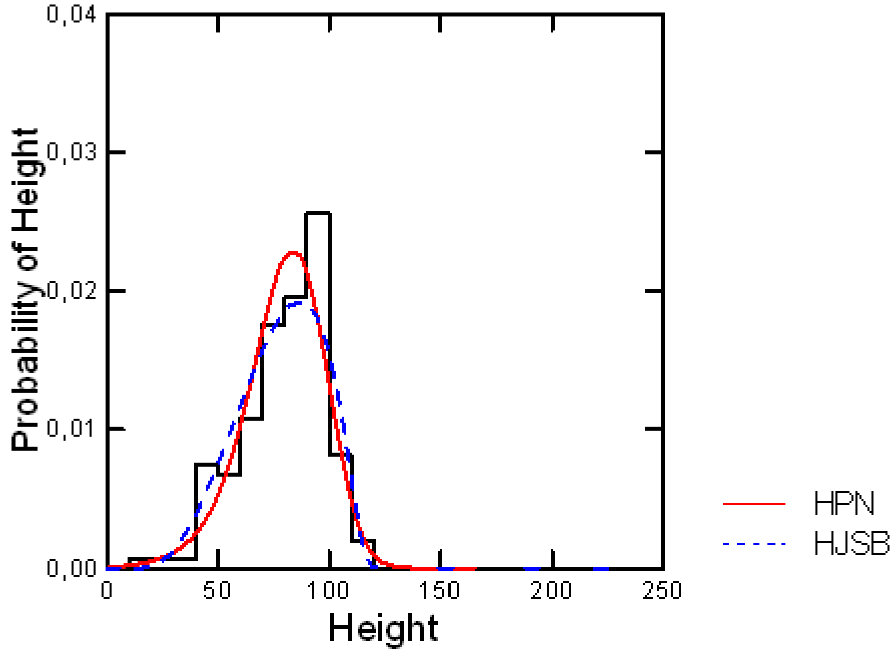


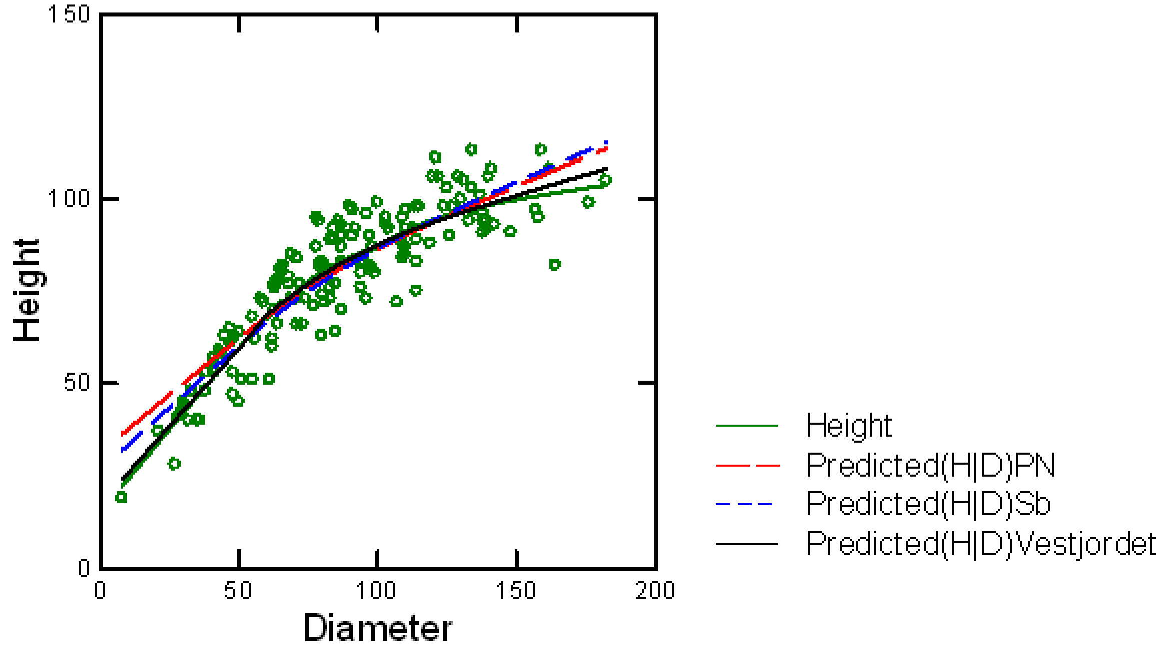




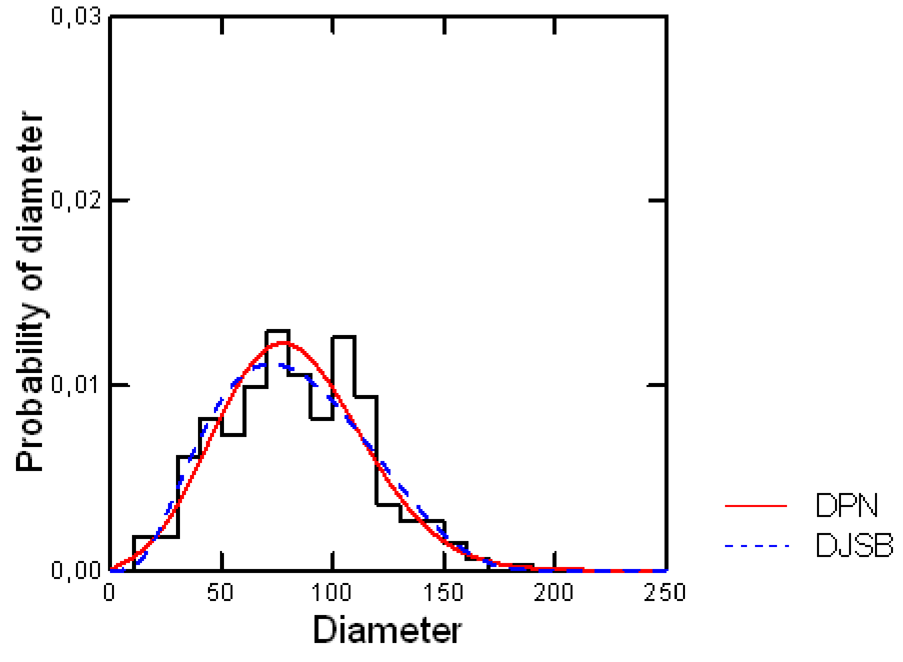
Results (↓) for STAND = 92

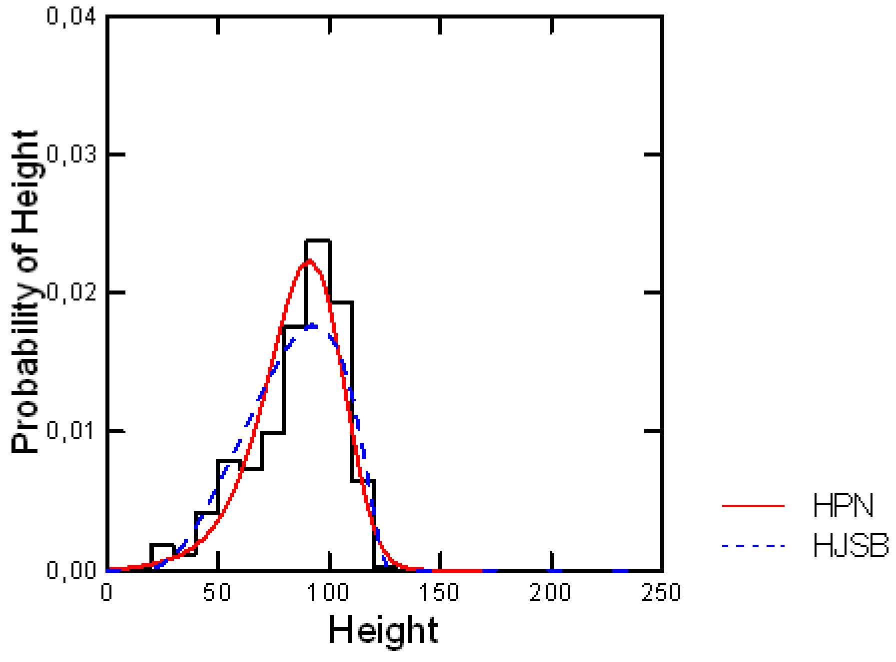


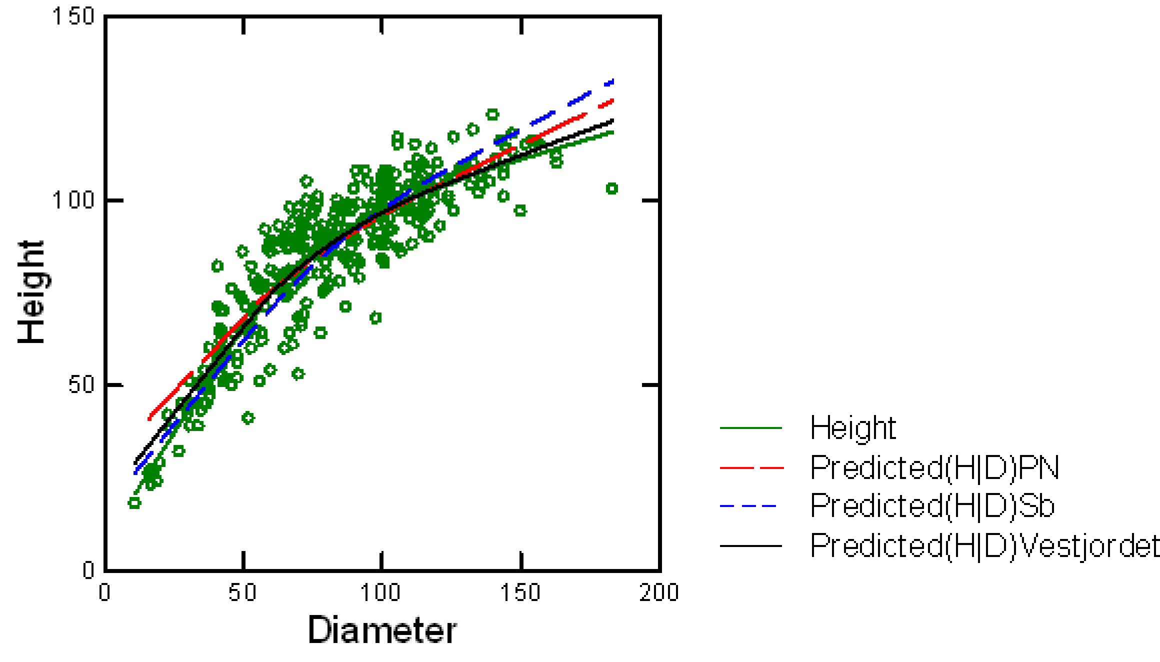




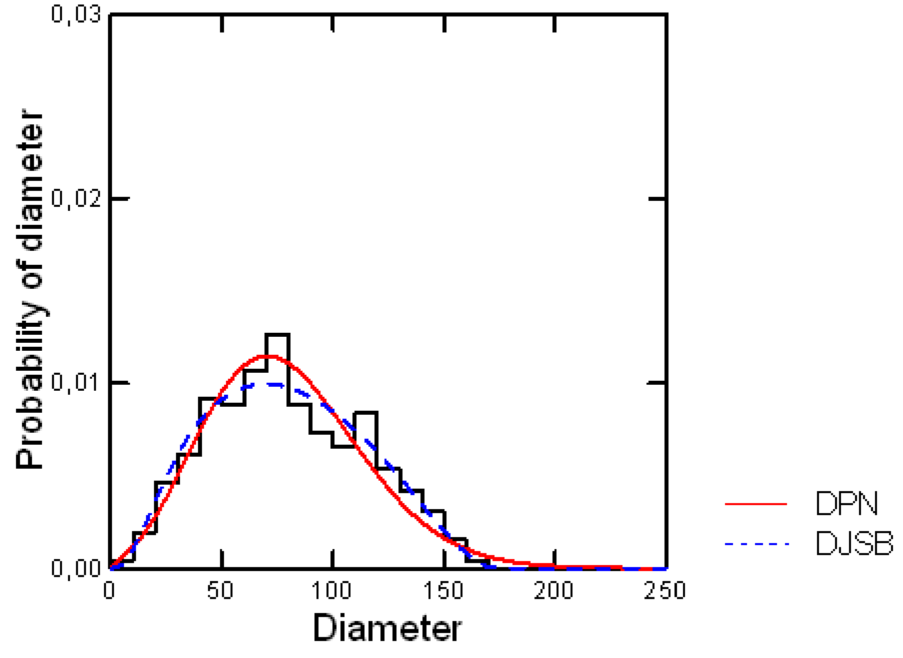
Results (↓) for STAND = 93

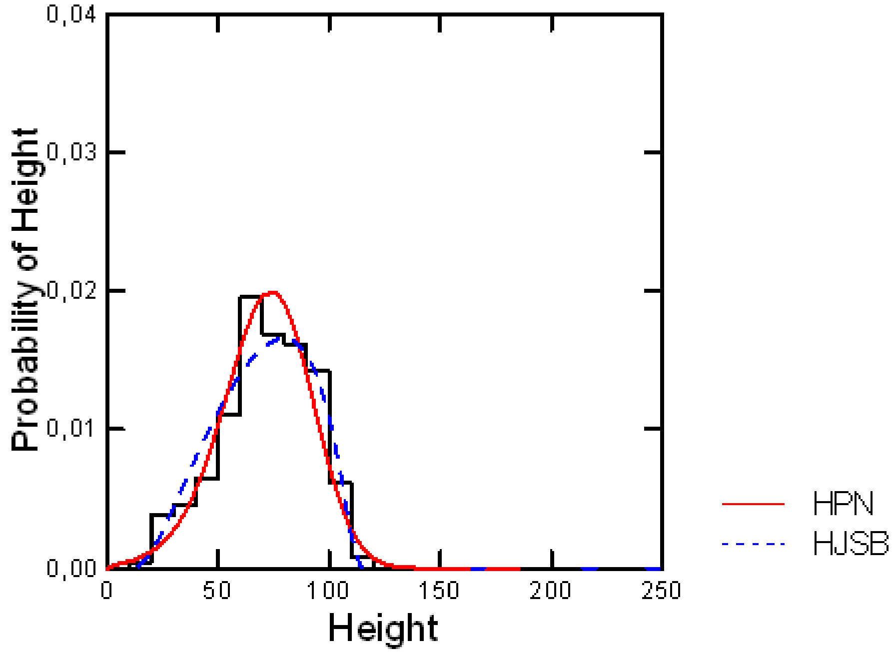


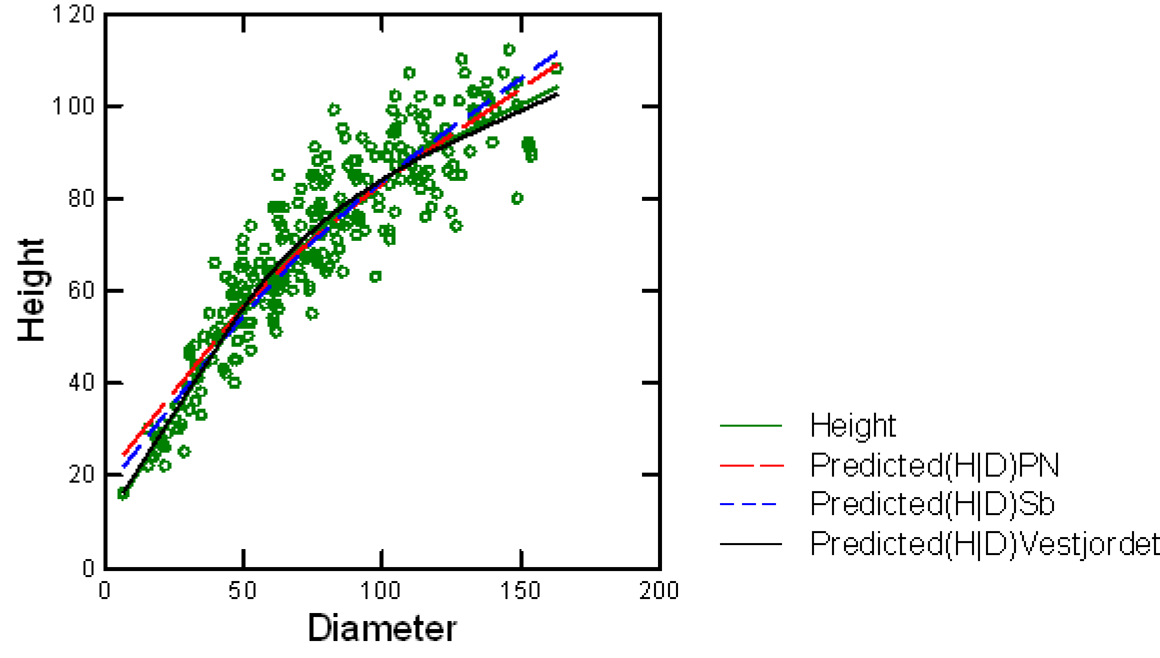




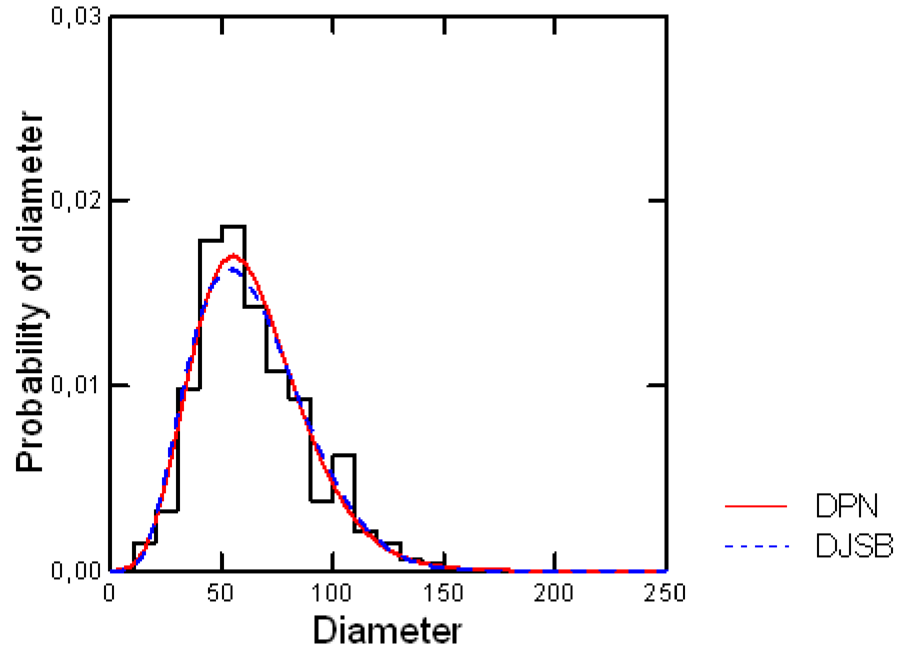
Results (↓) for STAND = 94

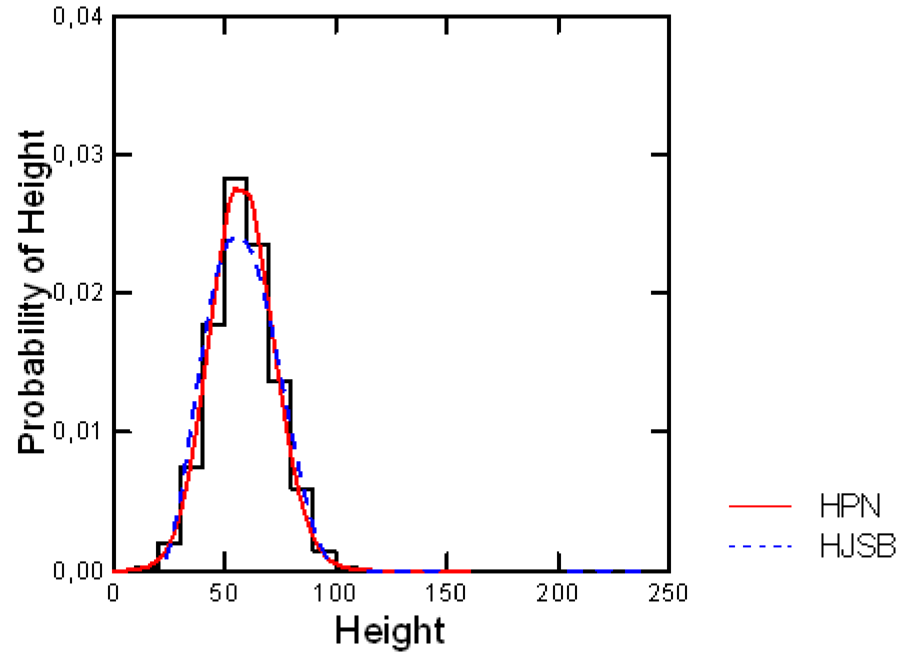


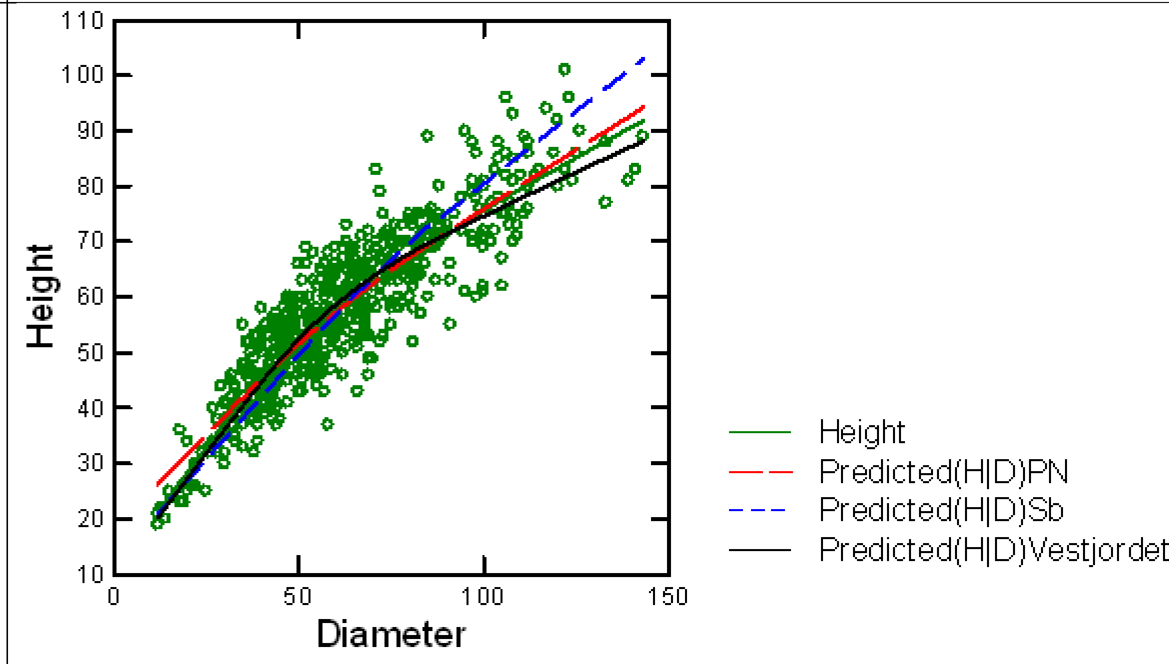




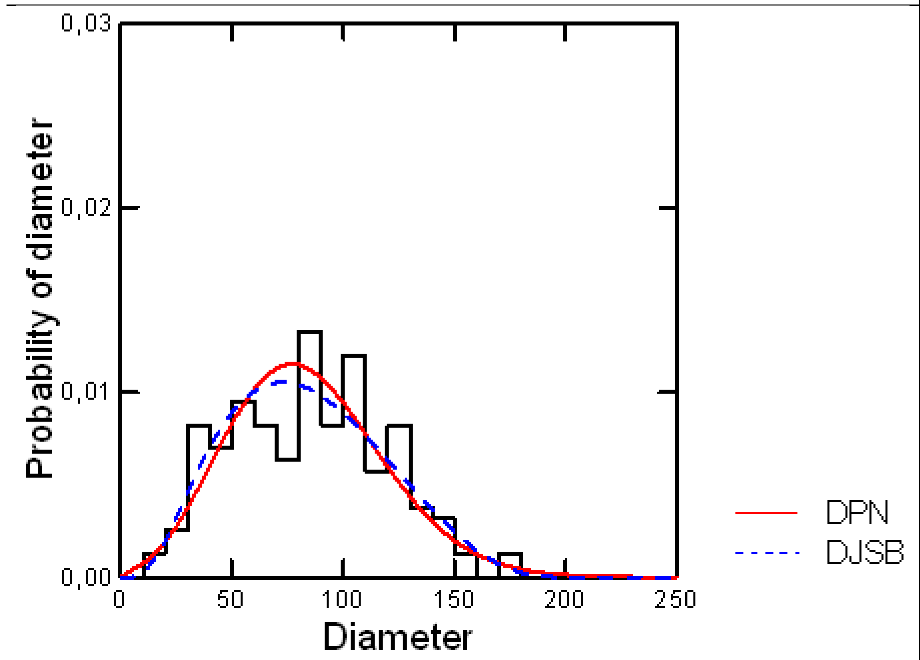
Results (↓) for STAND = 95

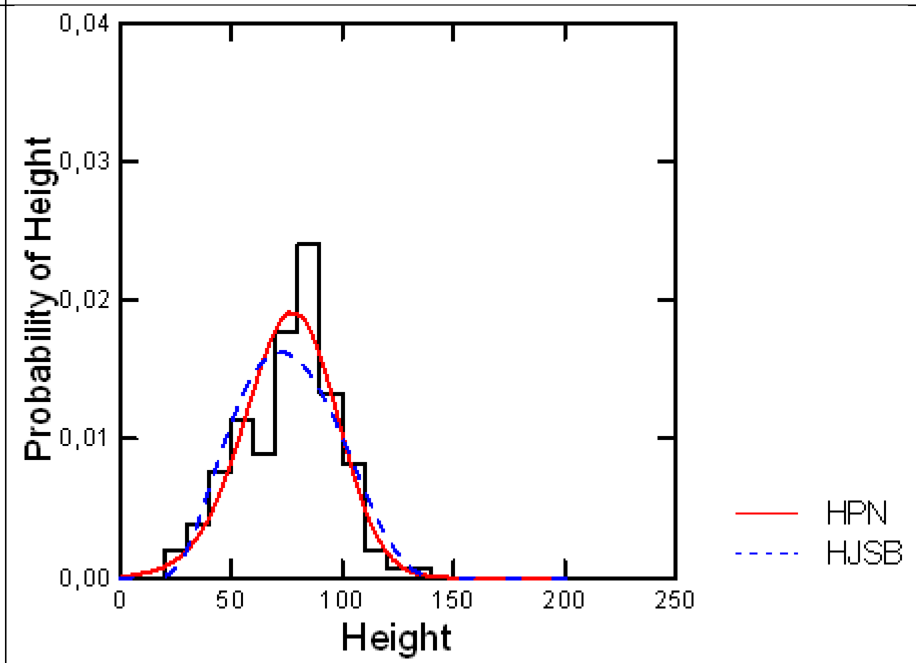


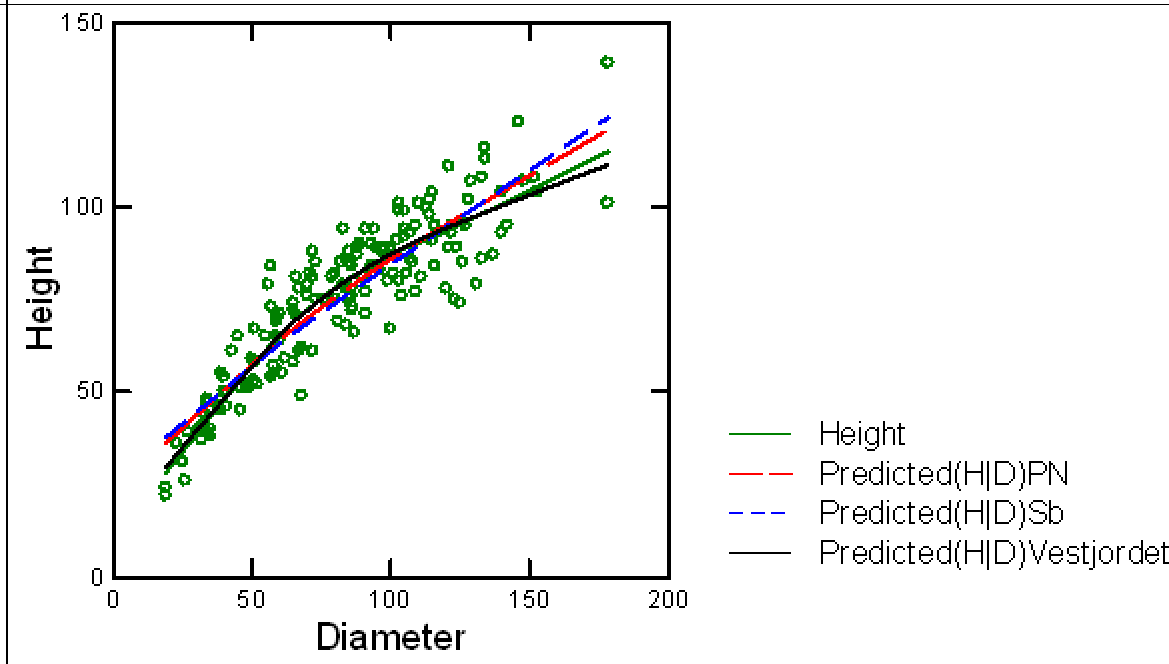




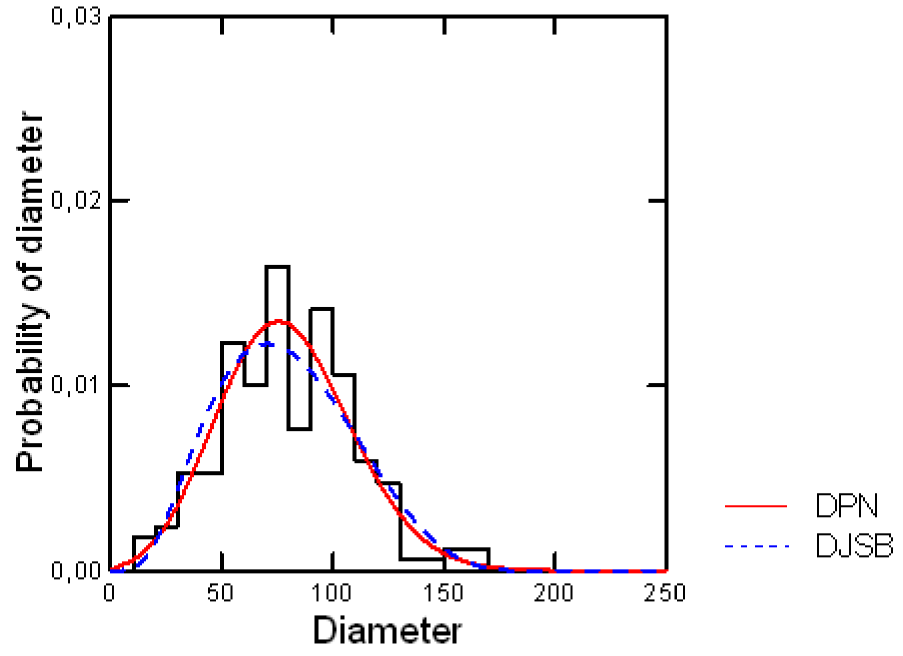
Results (↓) for STAND = 96

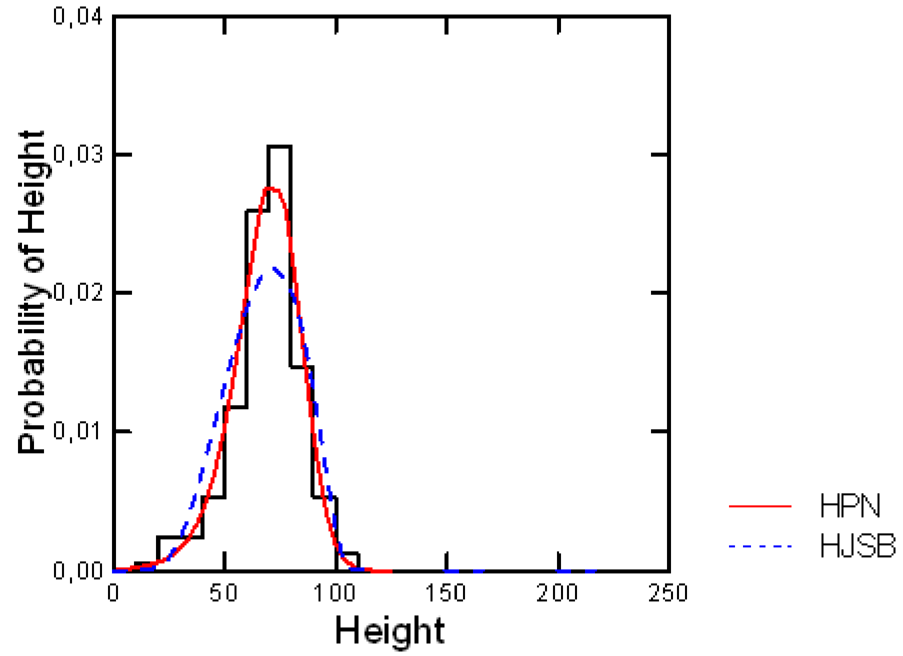


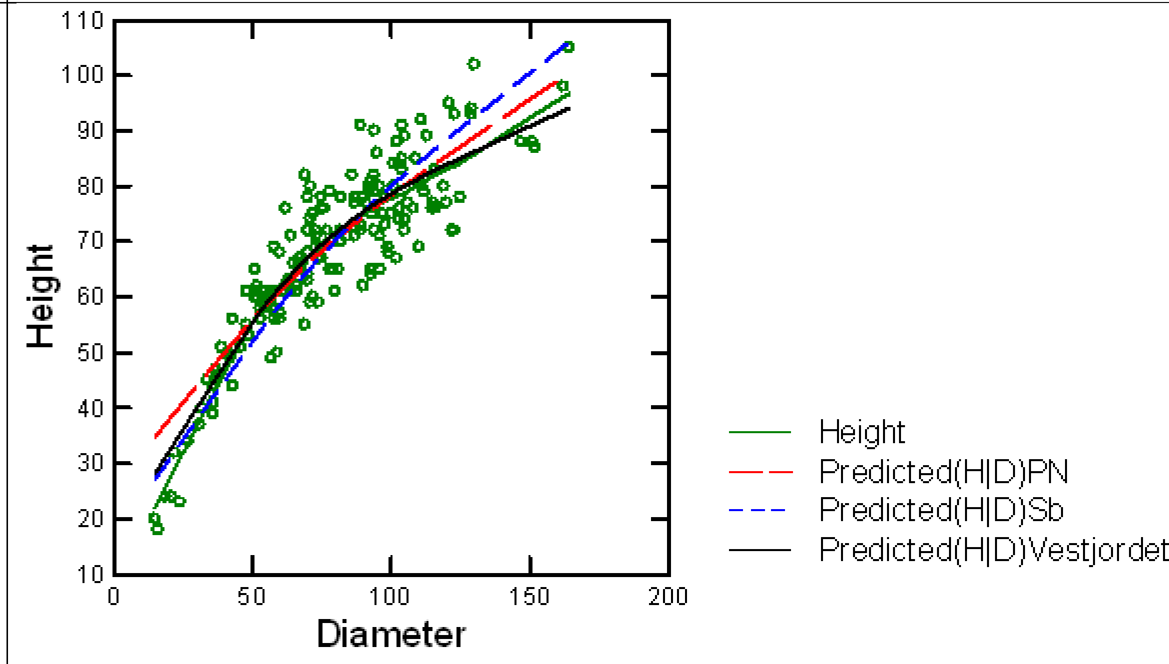




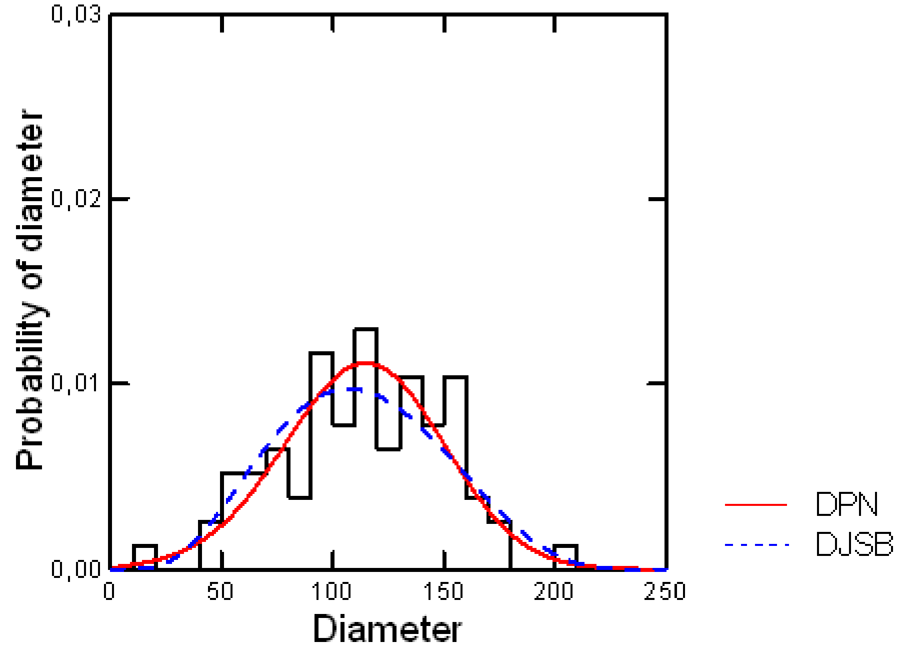
Results (↓) for STAND = 97

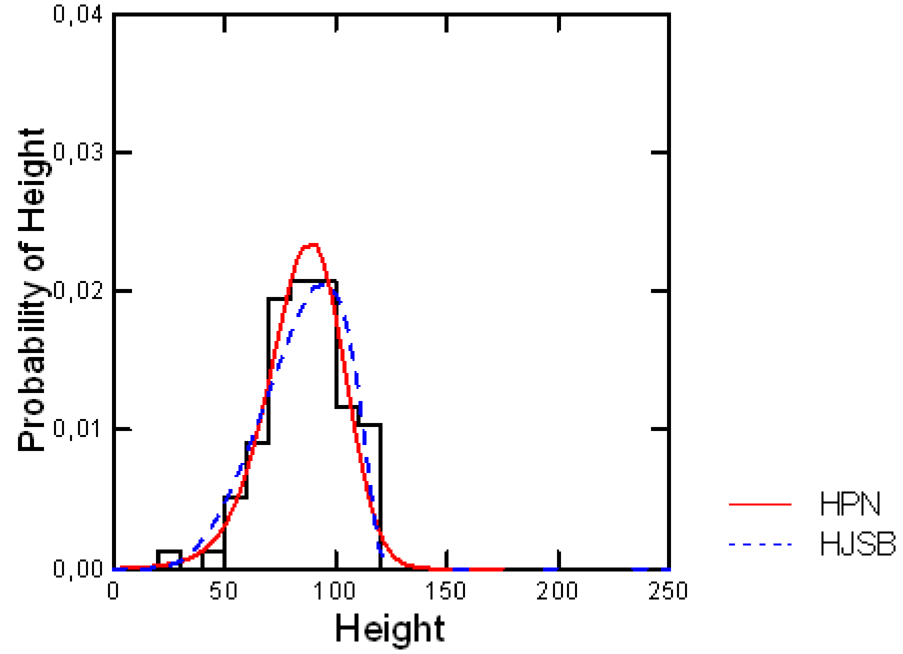


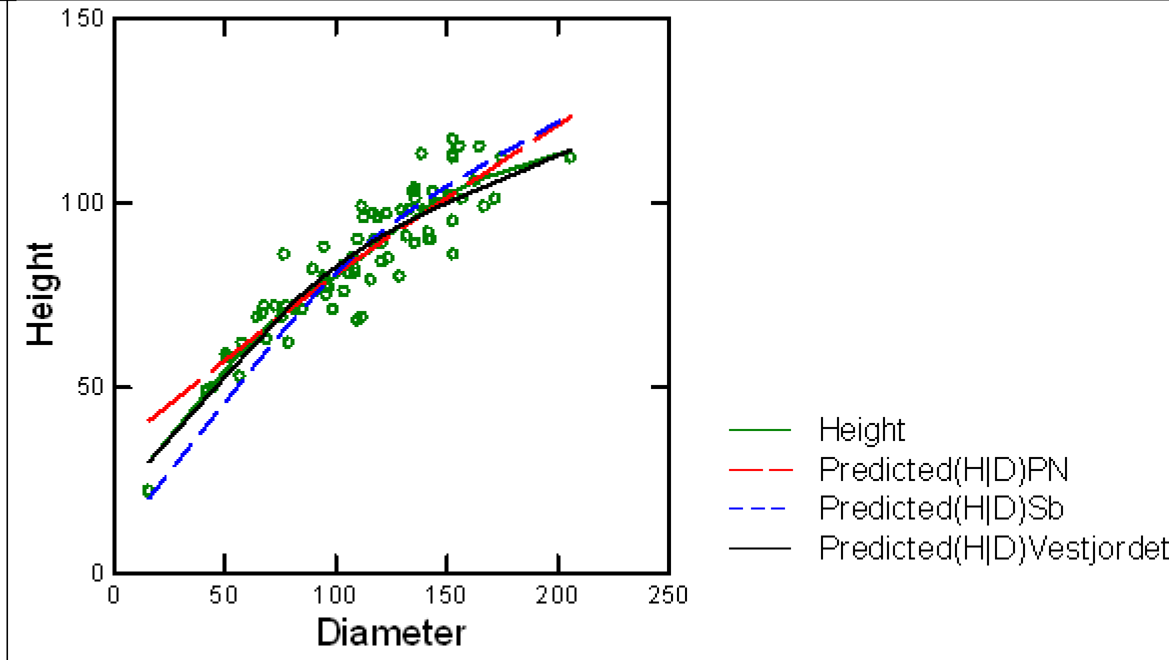




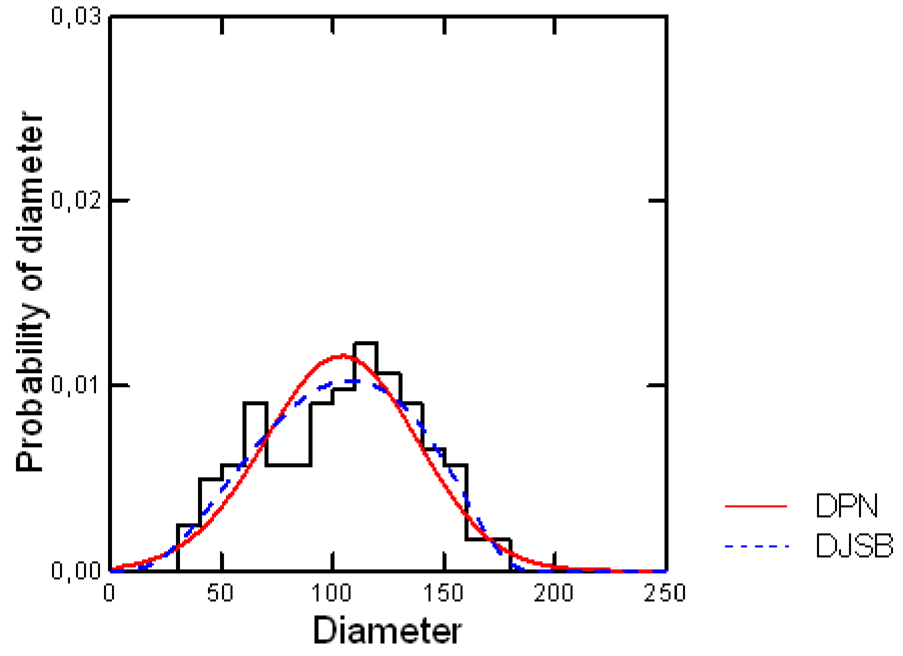
Results (↓) for STAND = 98

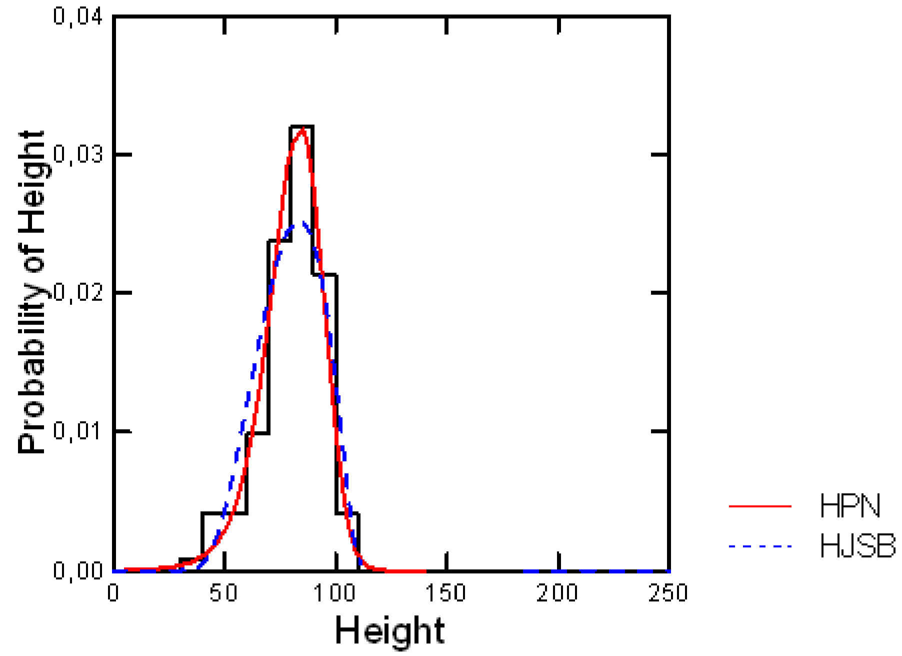


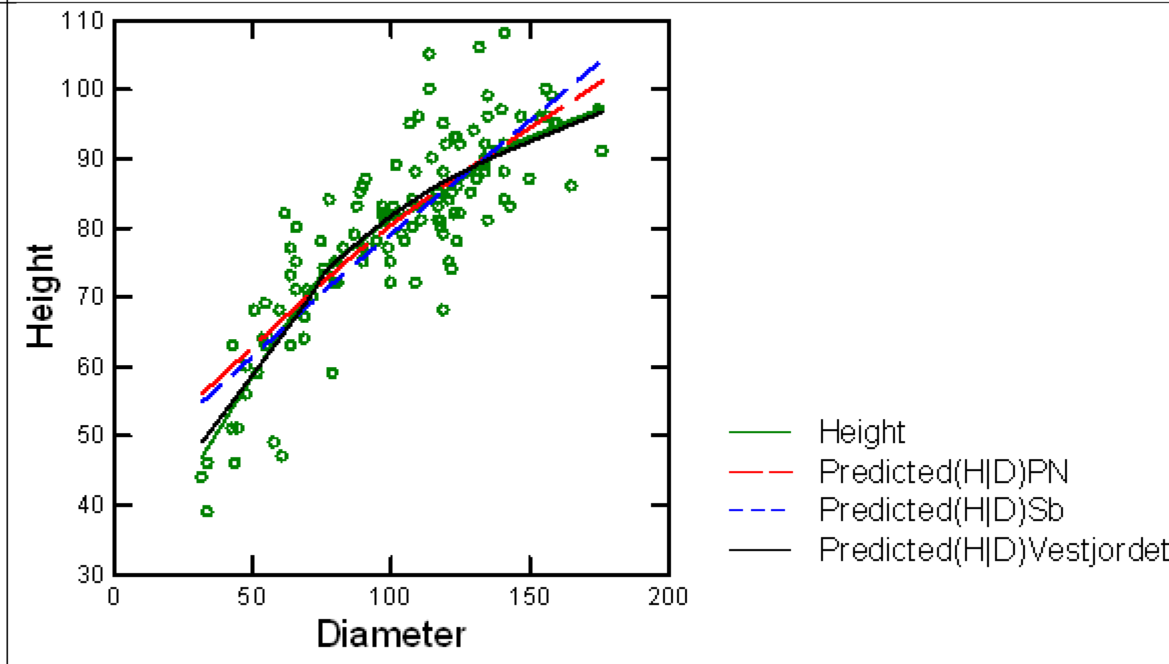




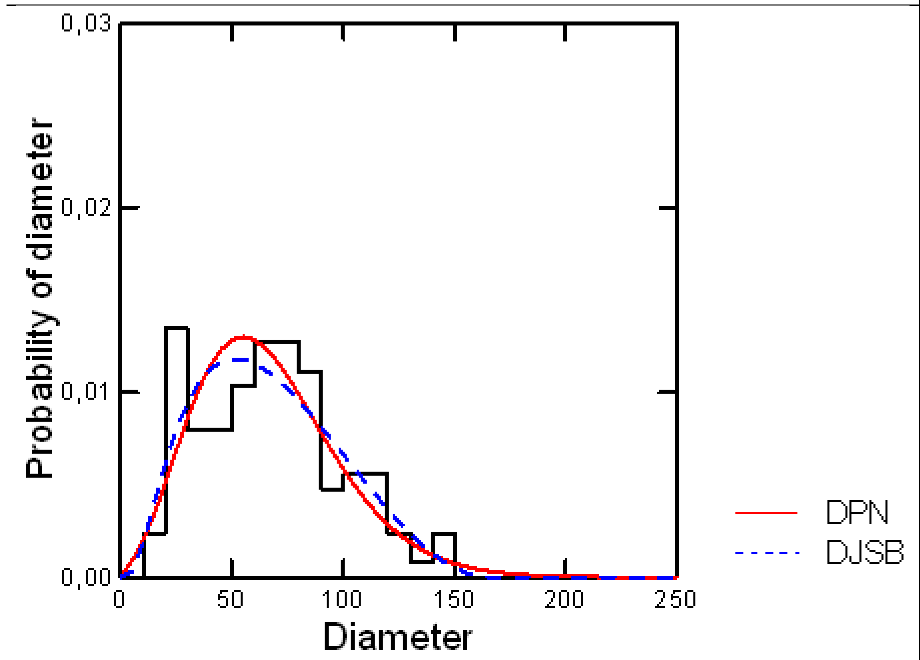
Results (↓) for STAND = 99

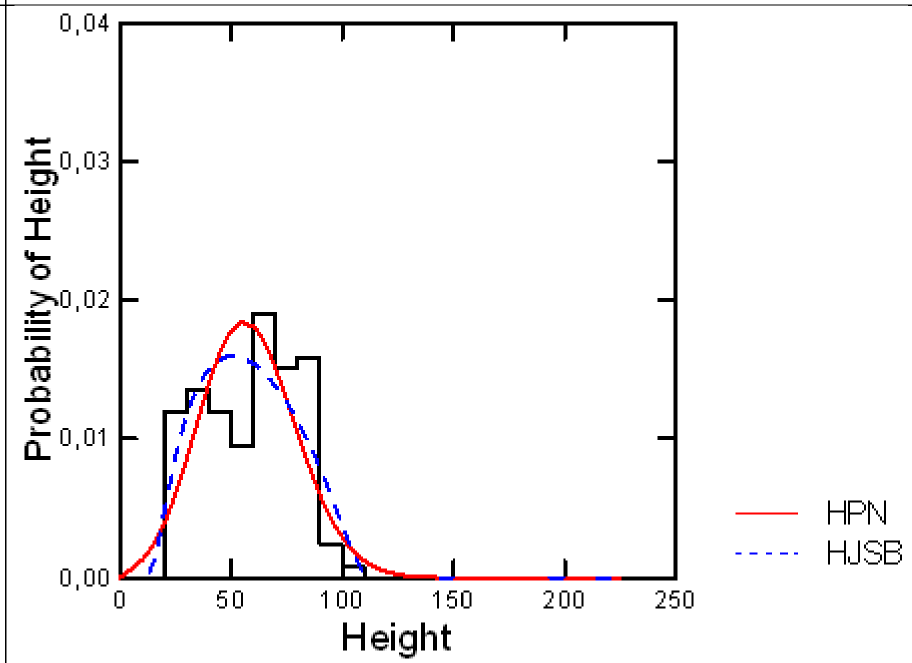


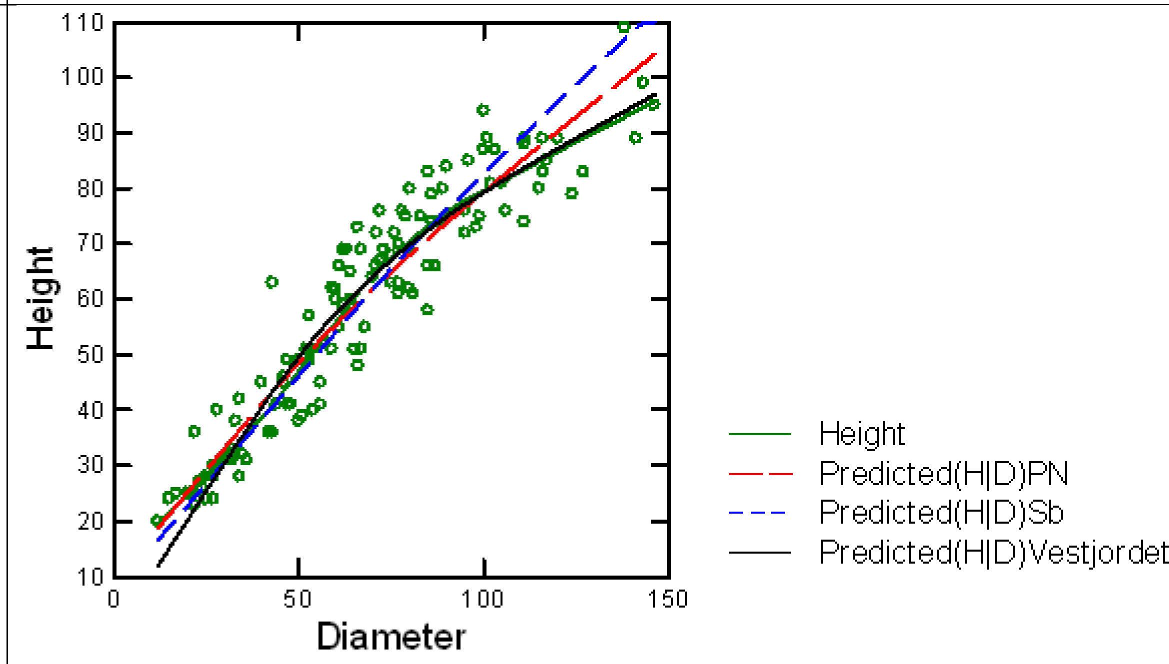




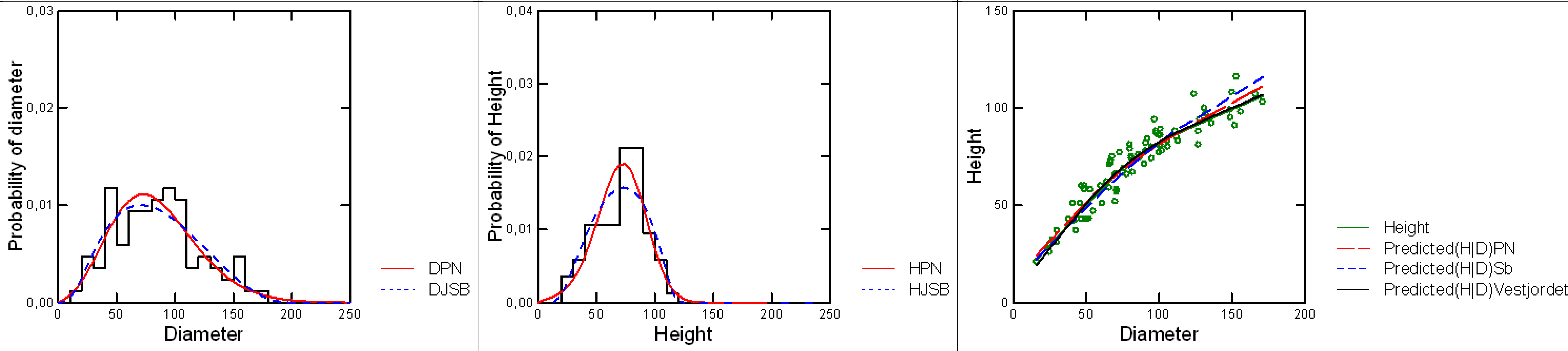
Results (↓) for STAND = 100



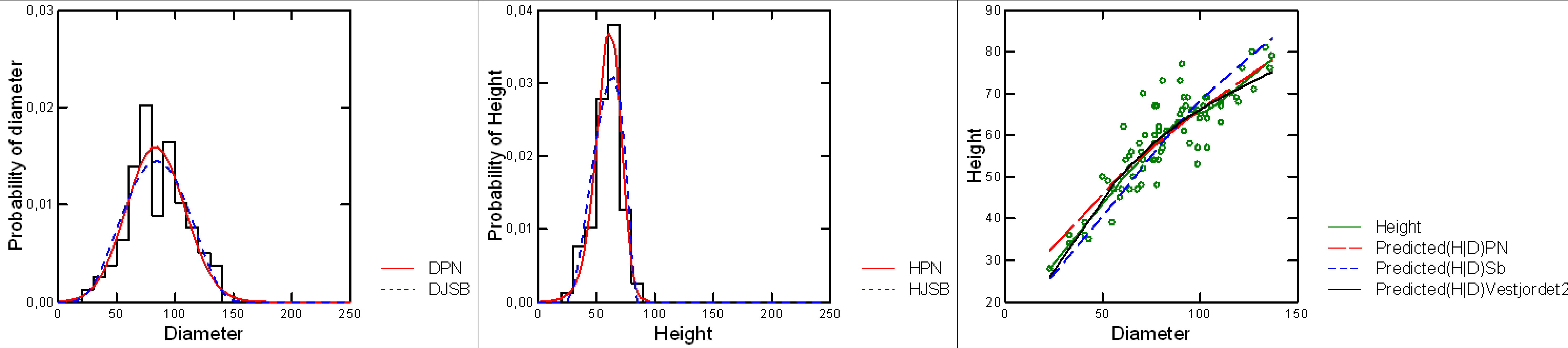




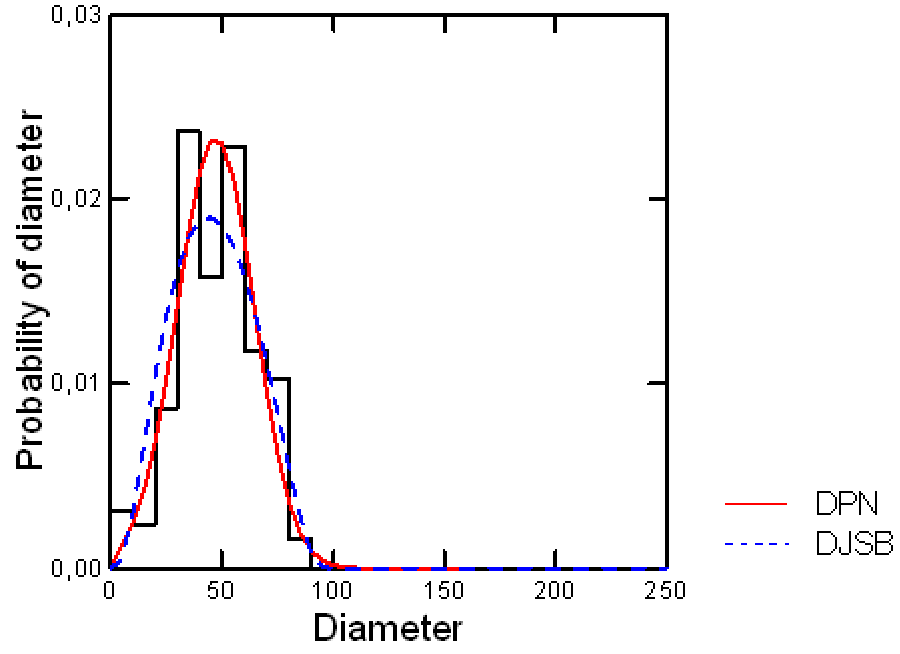
Results (↓) for STAND = 101

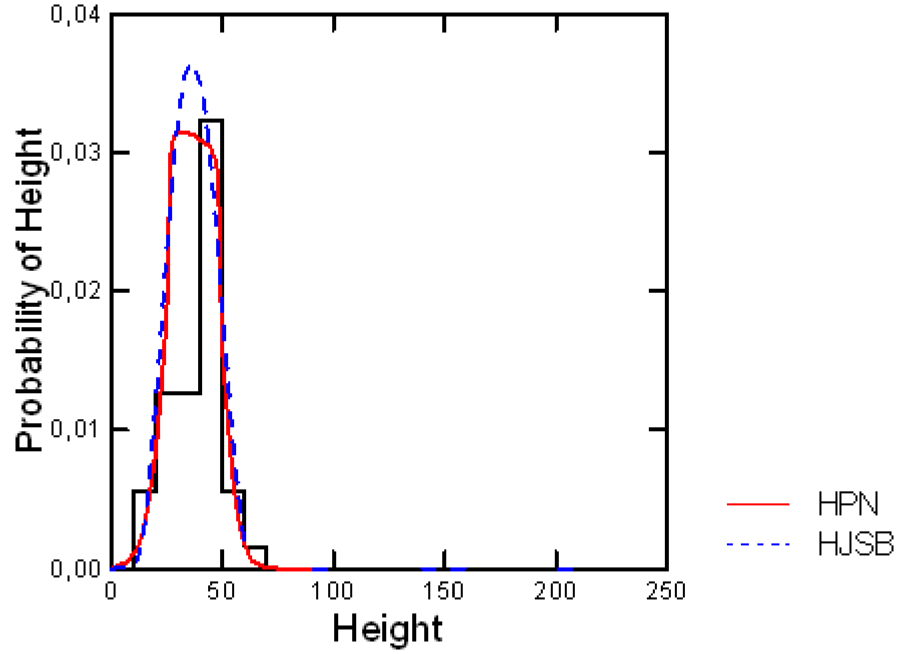


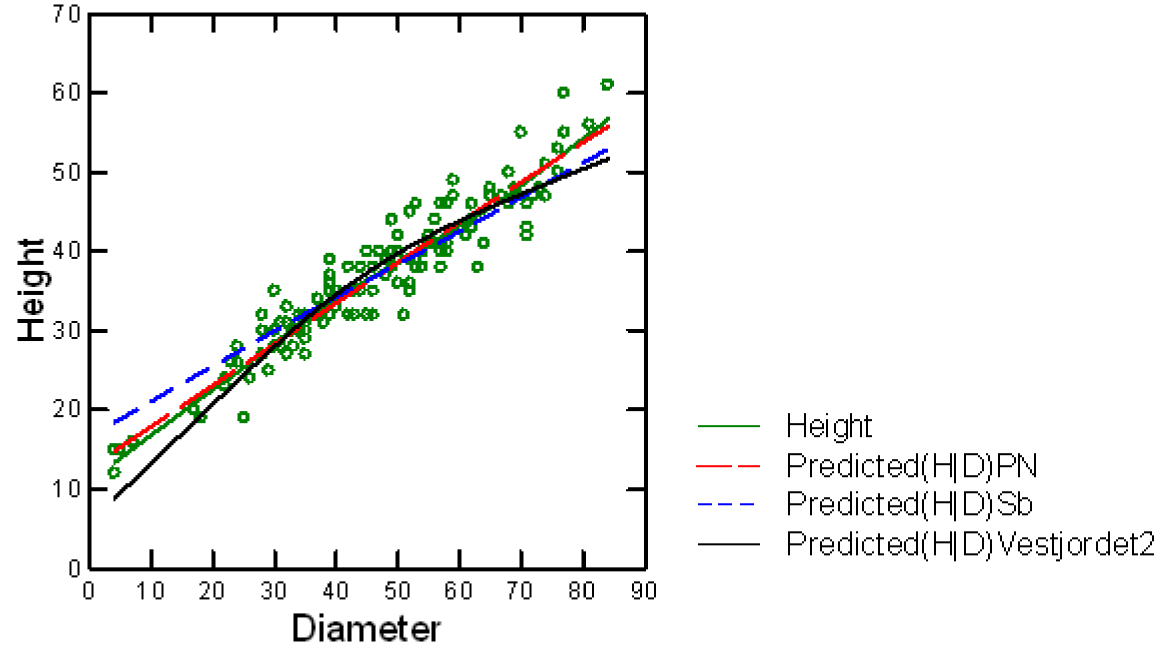
Results (↓) for STAND = 102



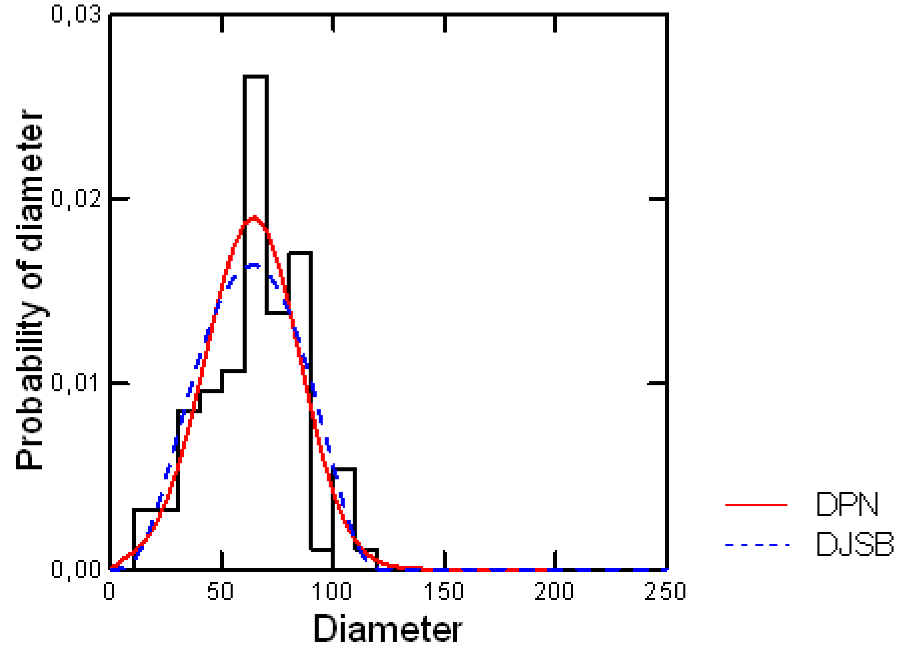
Results (↓) for STAND = 103

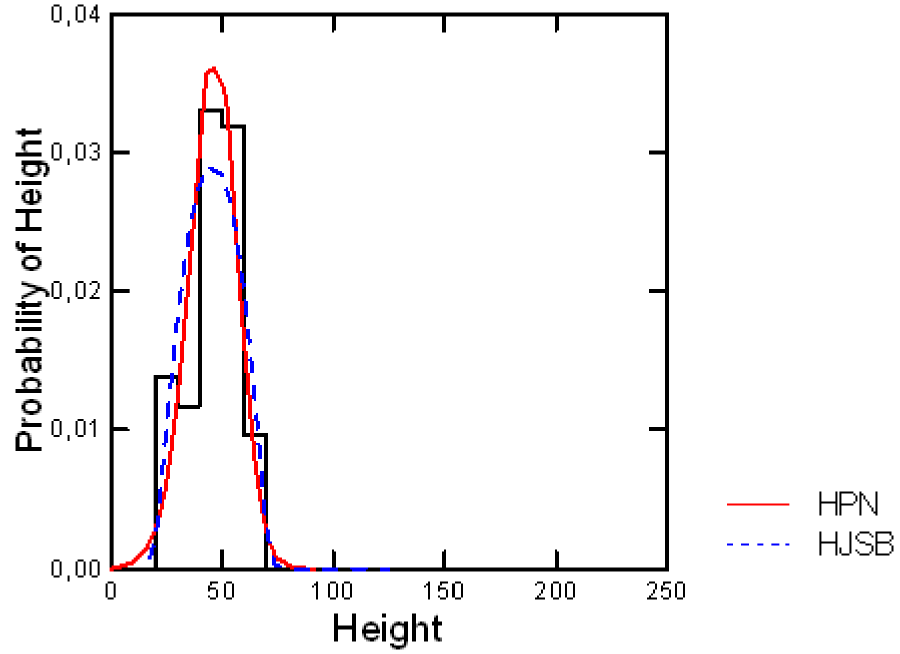


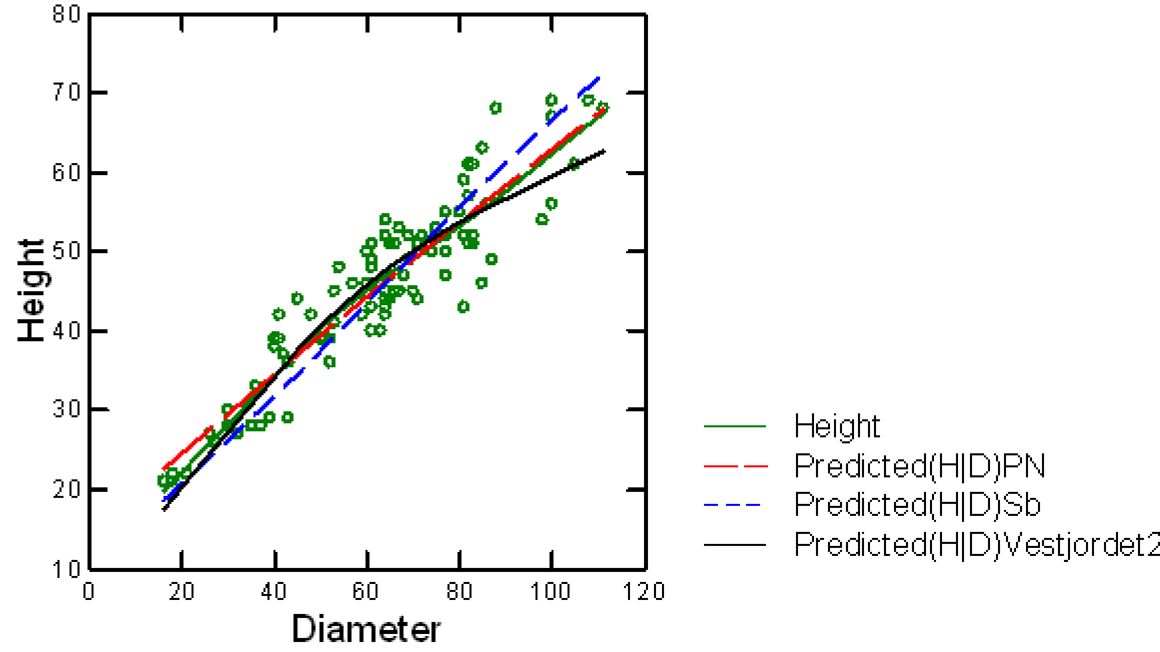




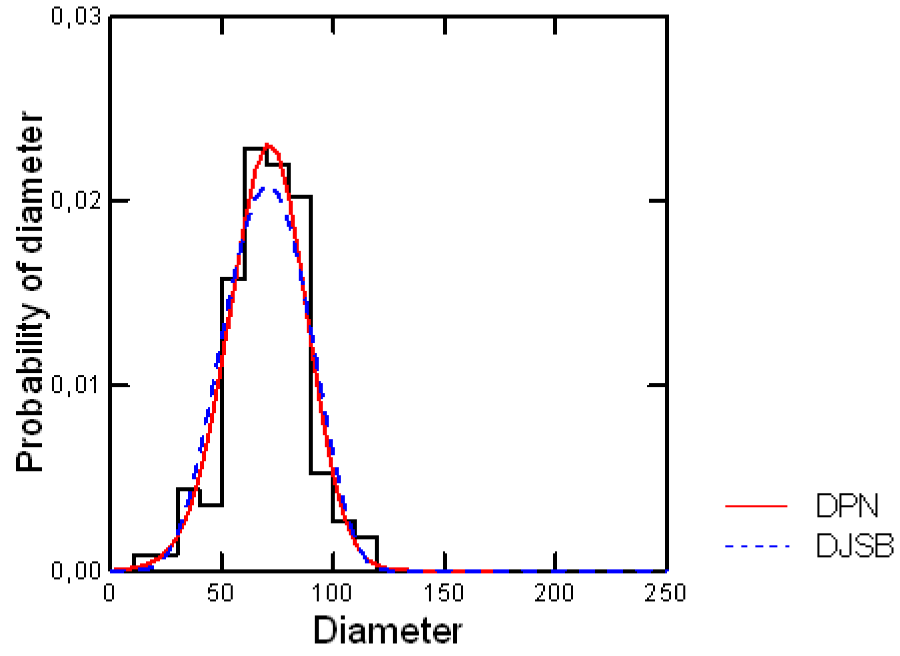
Results (↓) for STAND = 104

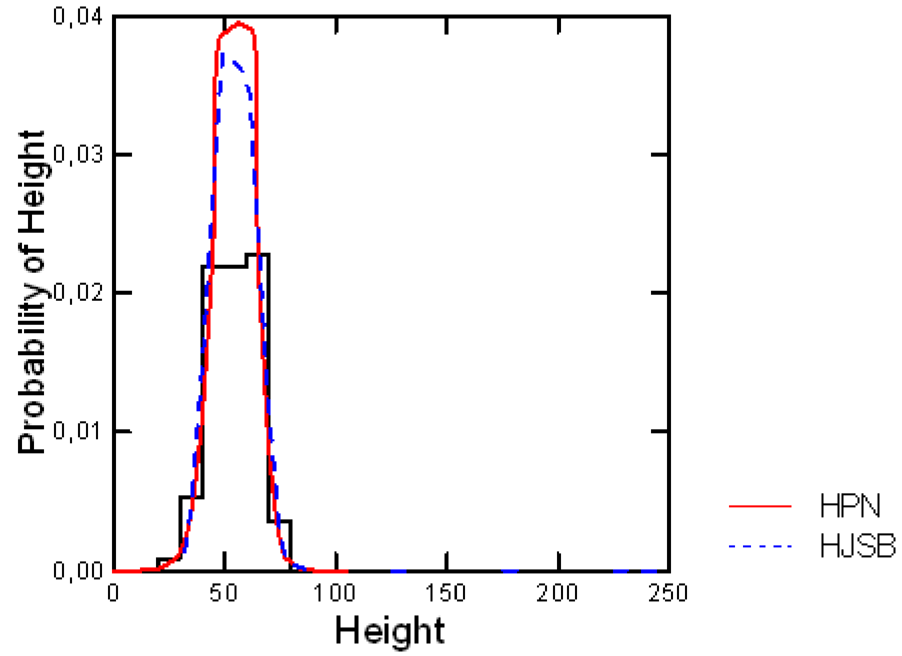


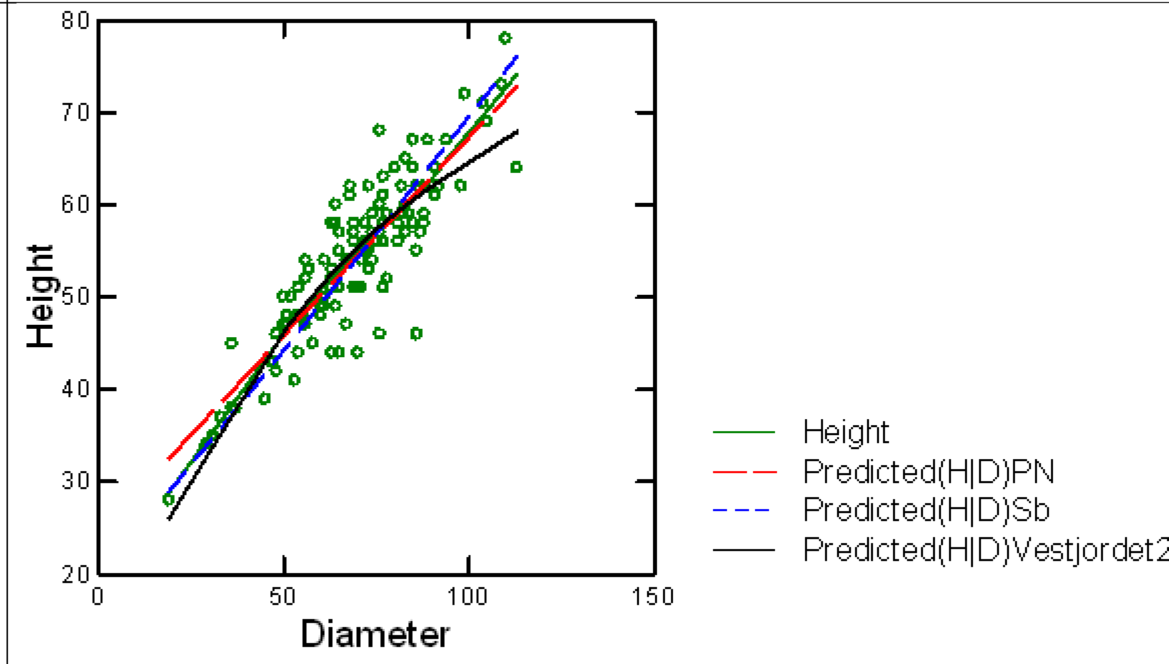




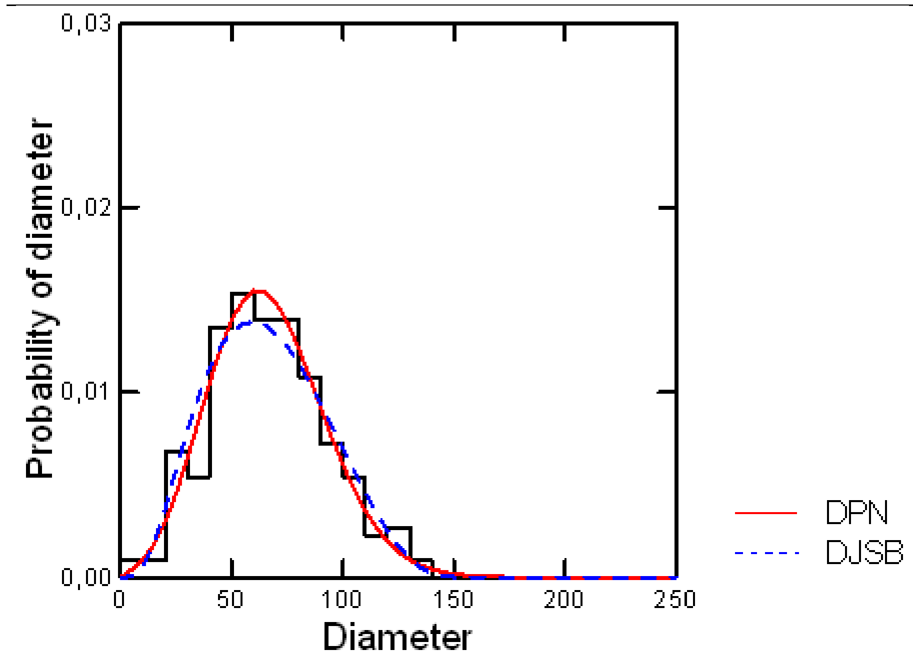
Results (↓) for STAND = 105

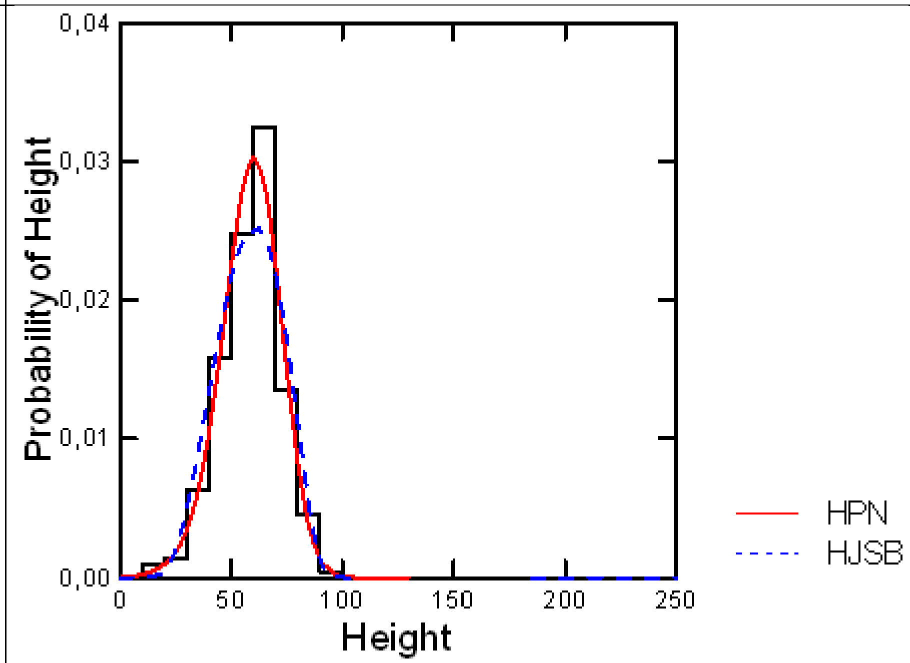


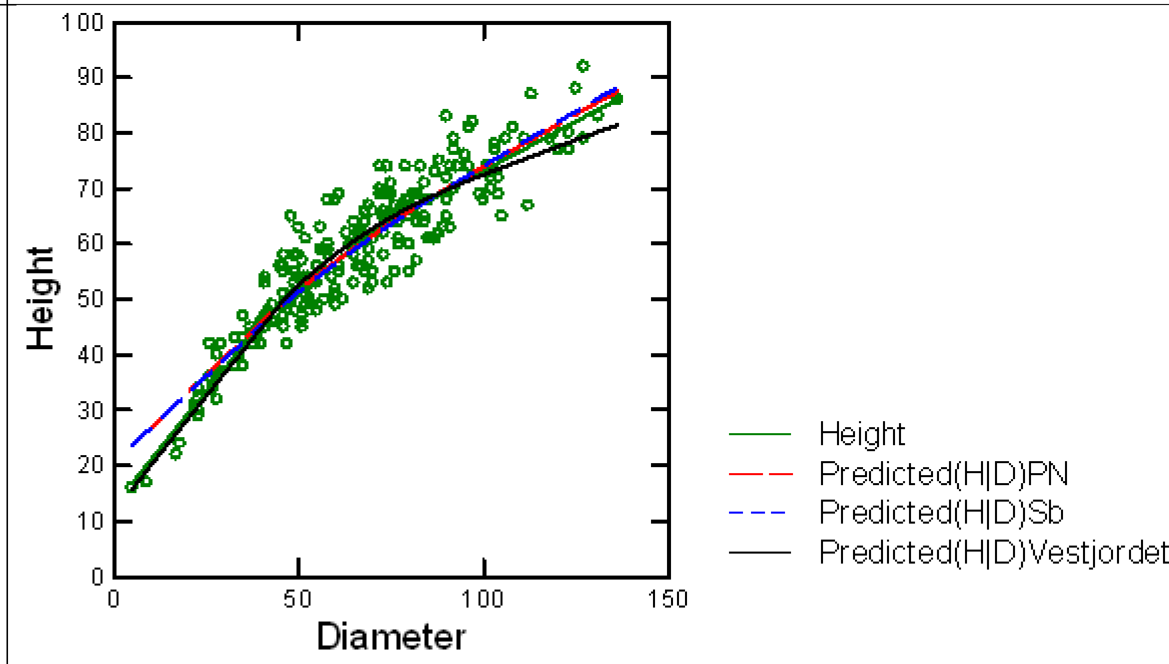




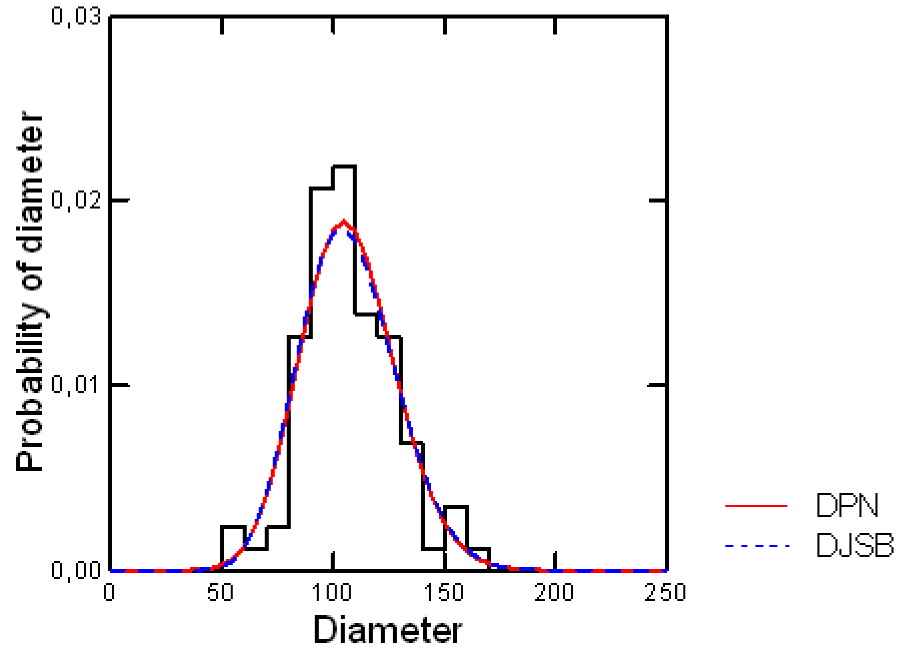
Results (↓) for STAND = 106

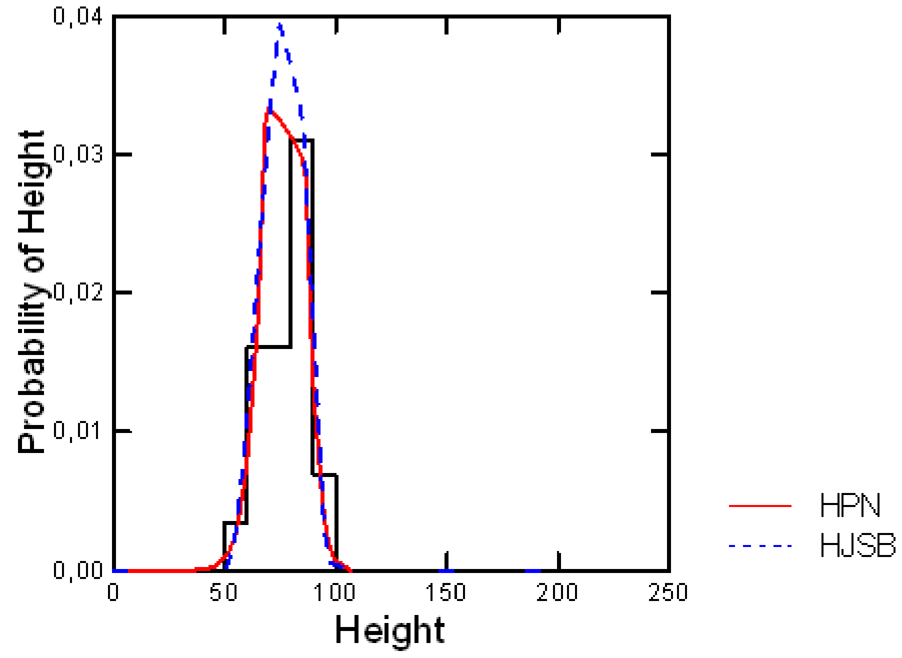


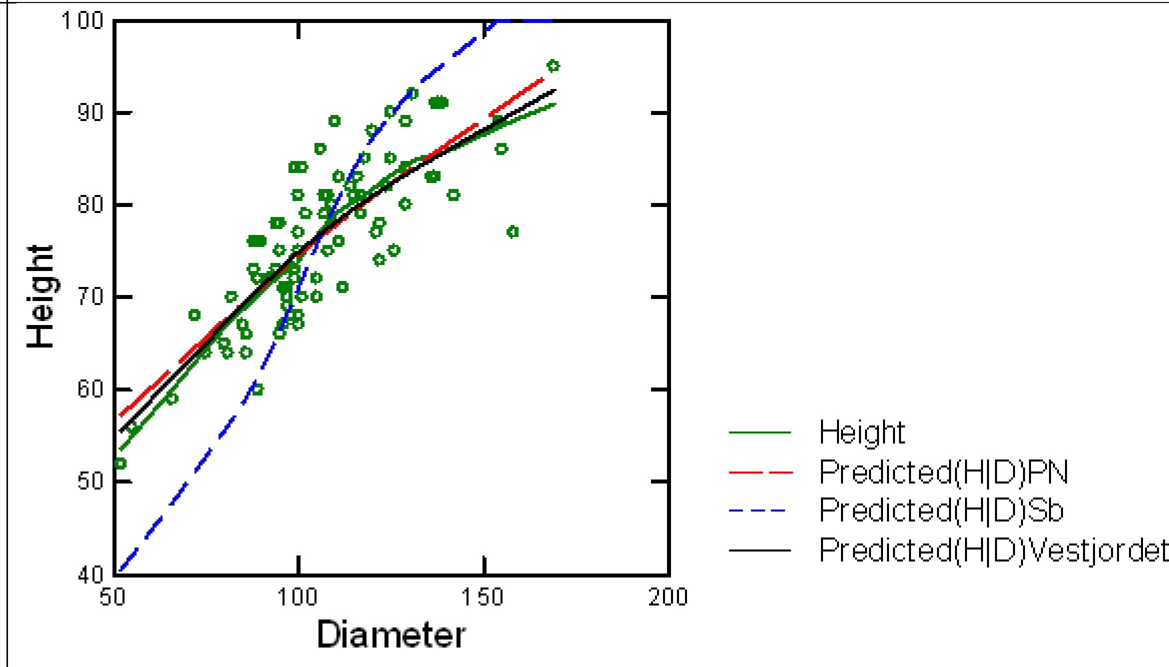




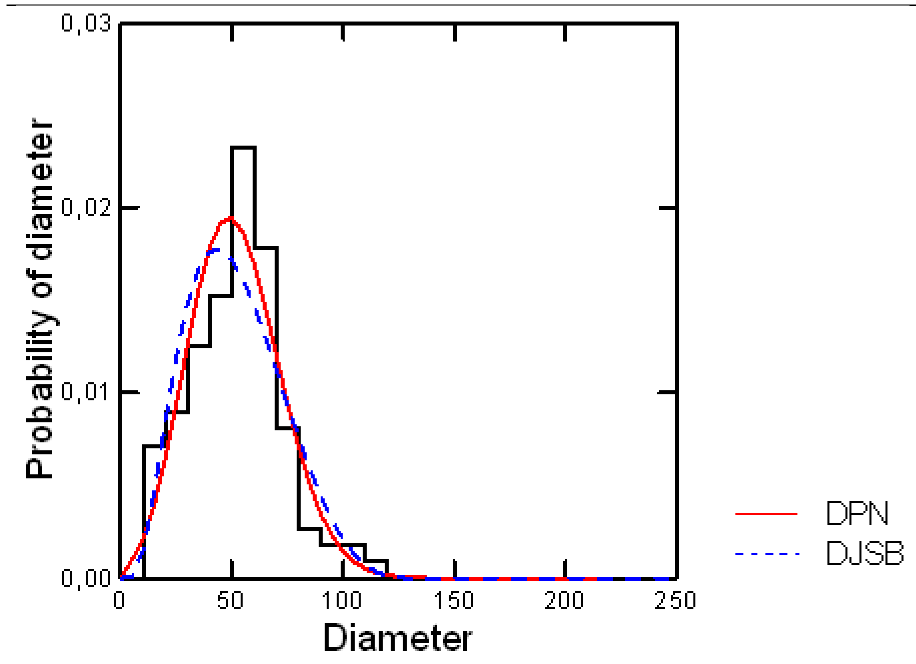
Results (↓) for STAND = 107

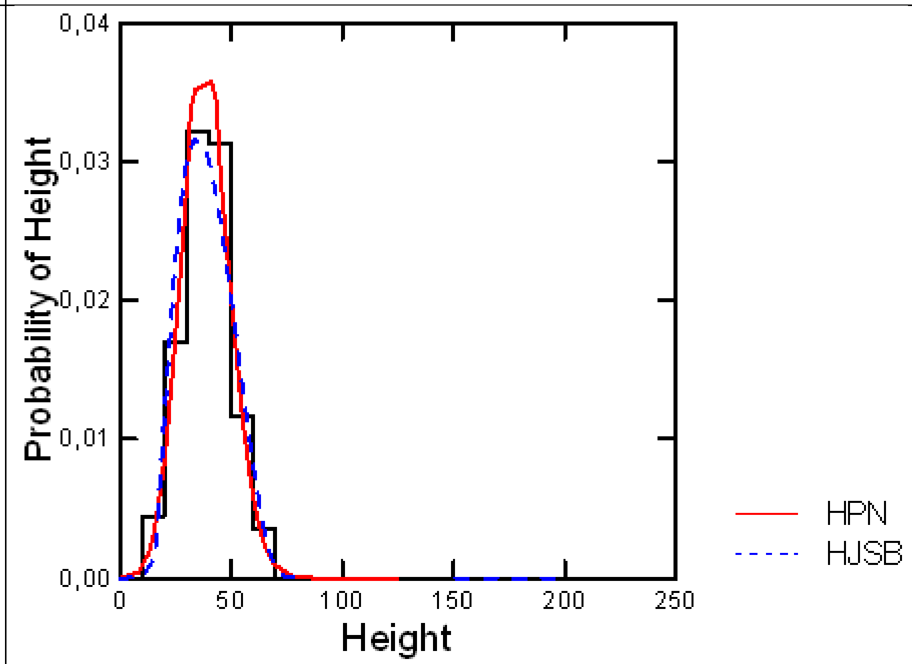


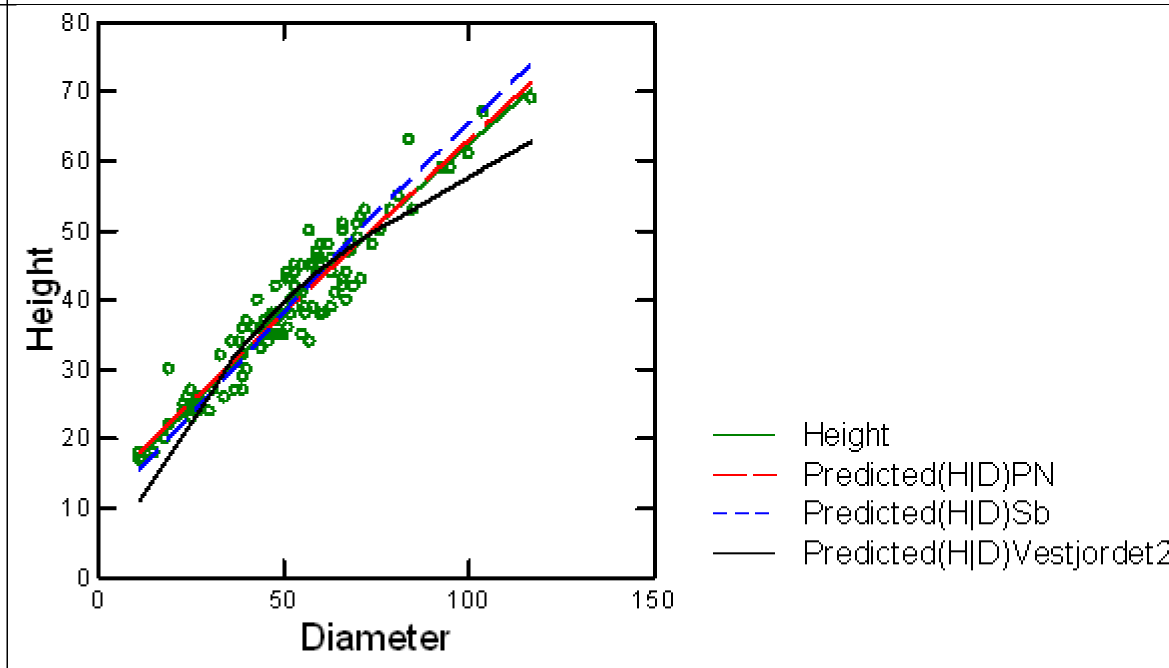




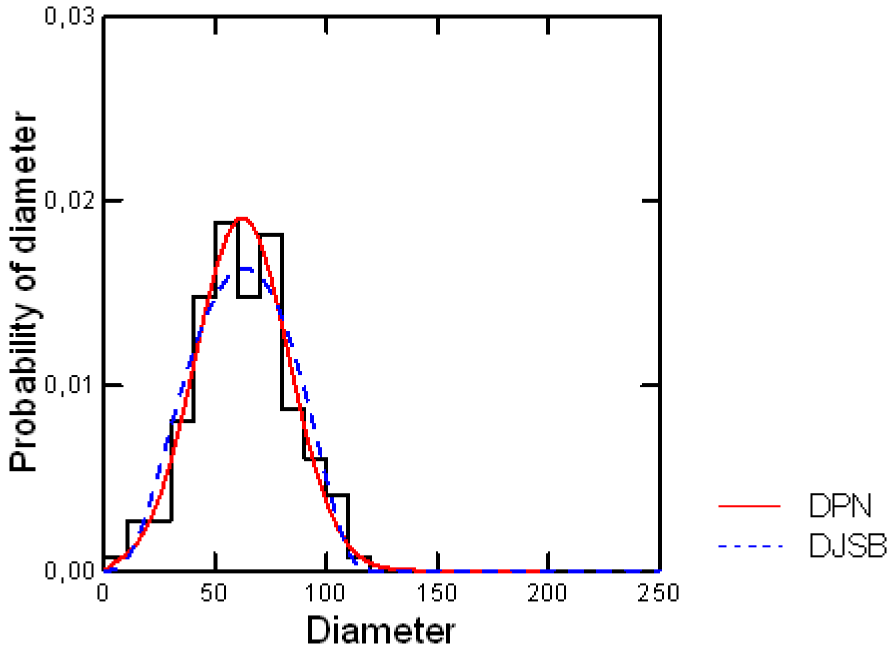
Results (↓) for STAND = 108

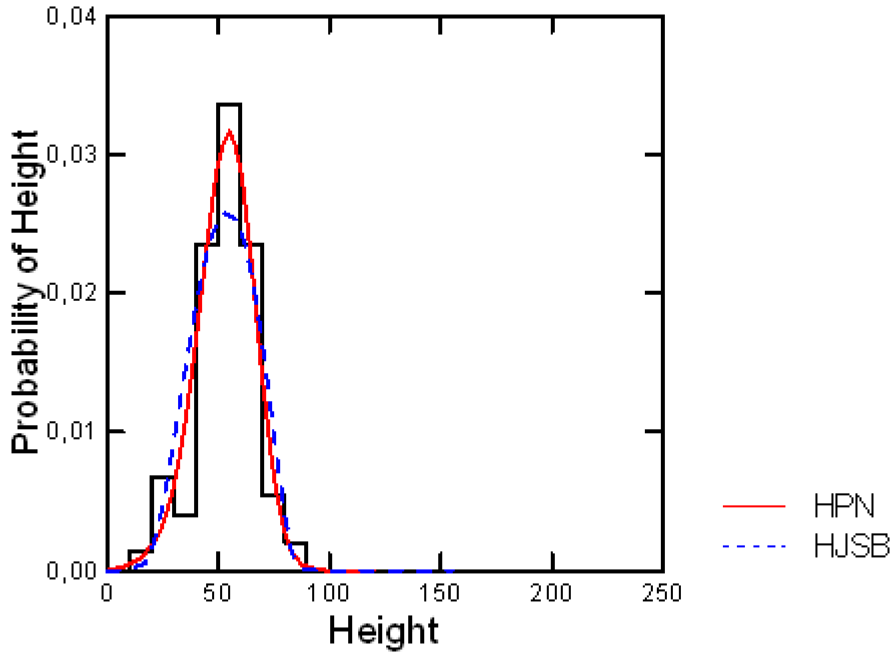


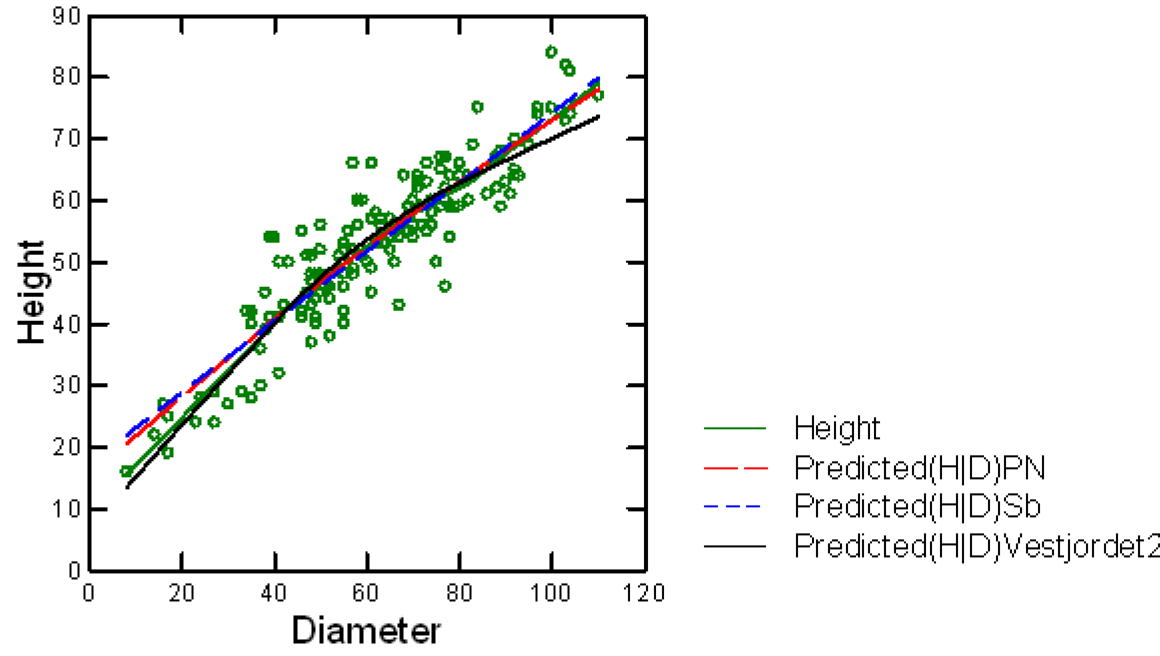




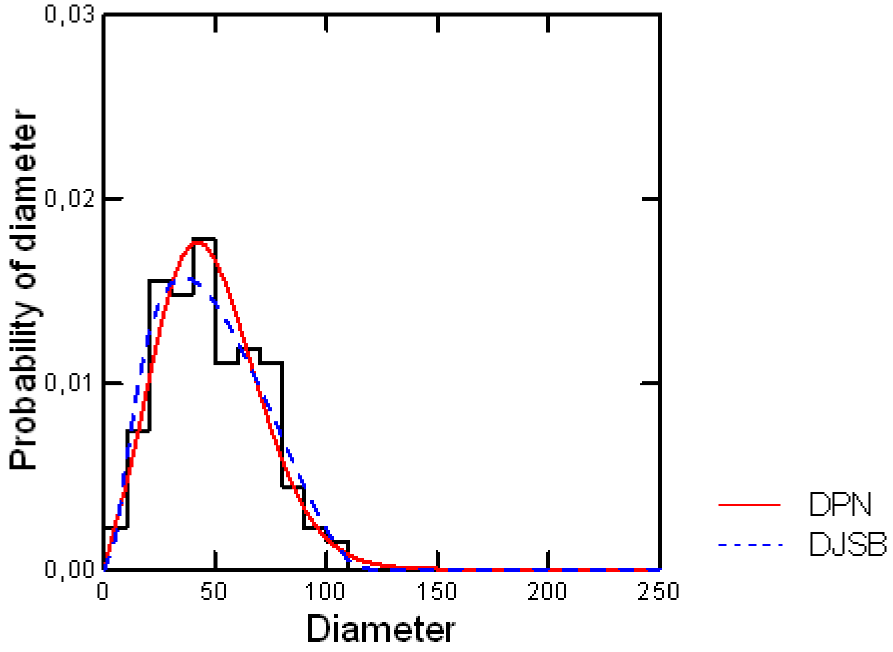
Results (↓) for STAND = 109

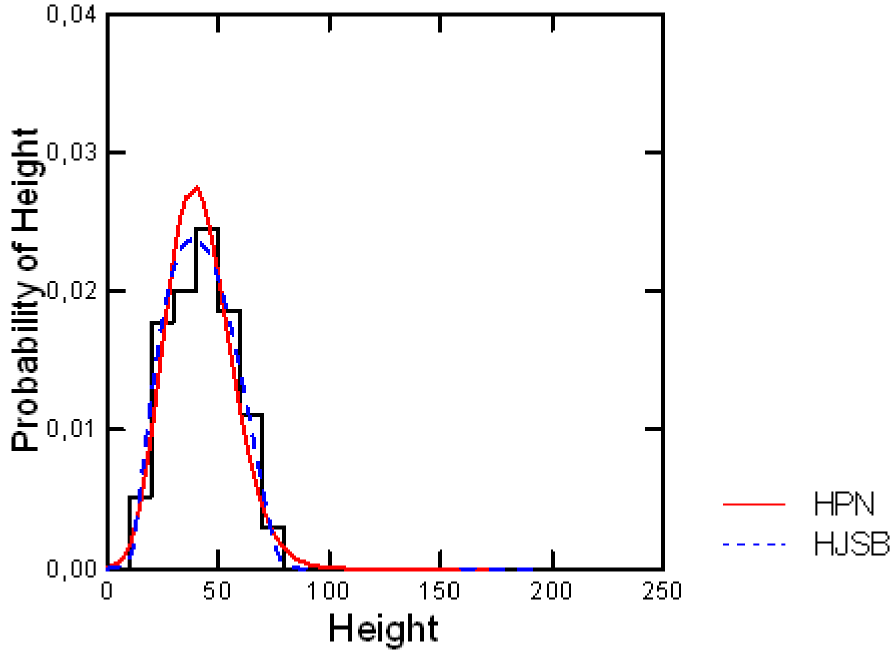


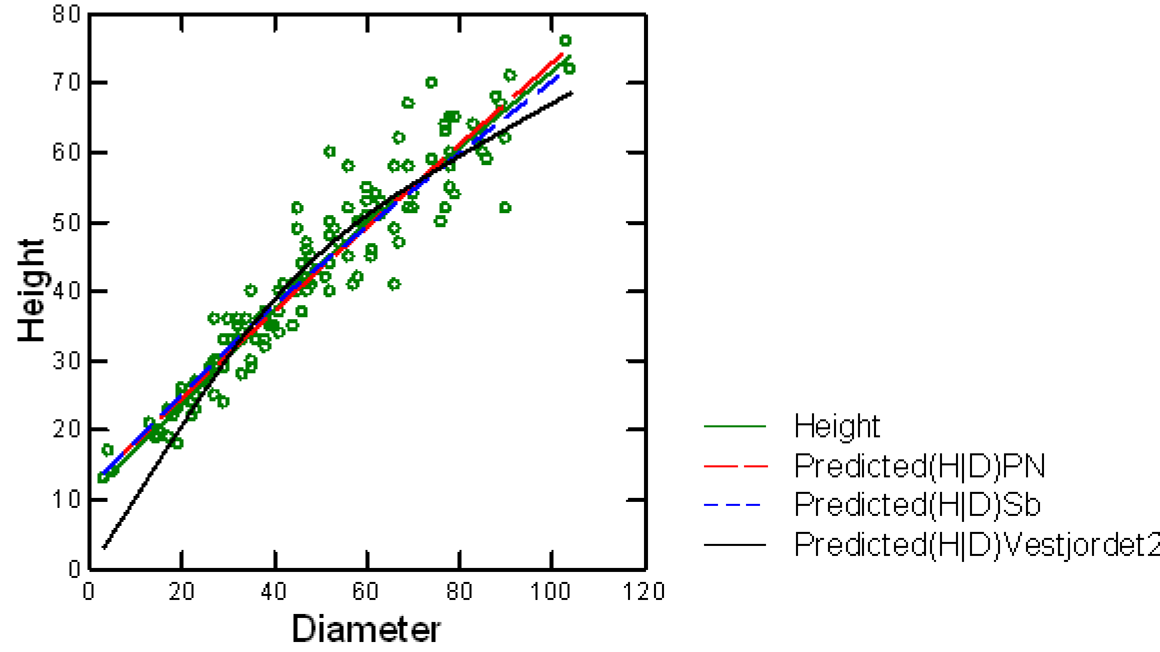




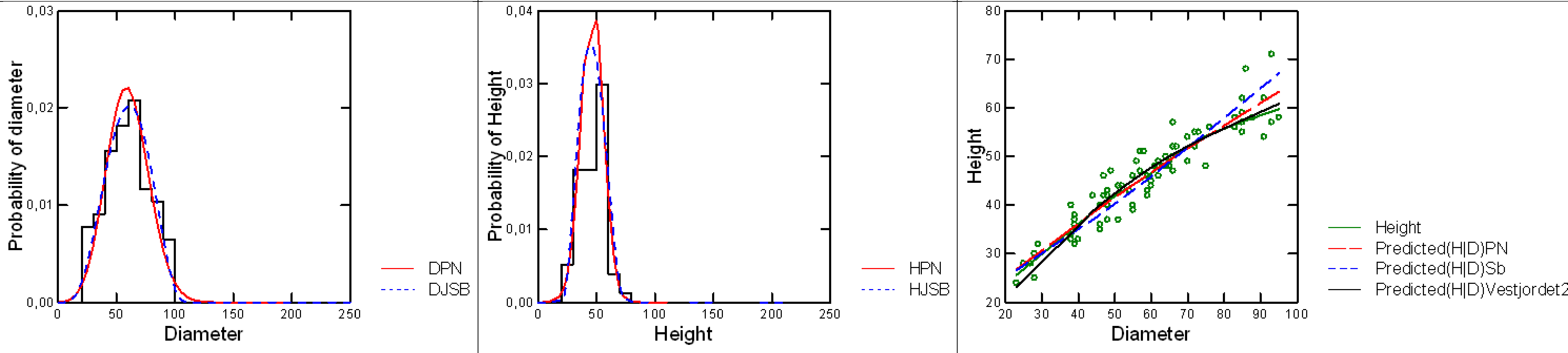
Results (↓) for STAND = 110



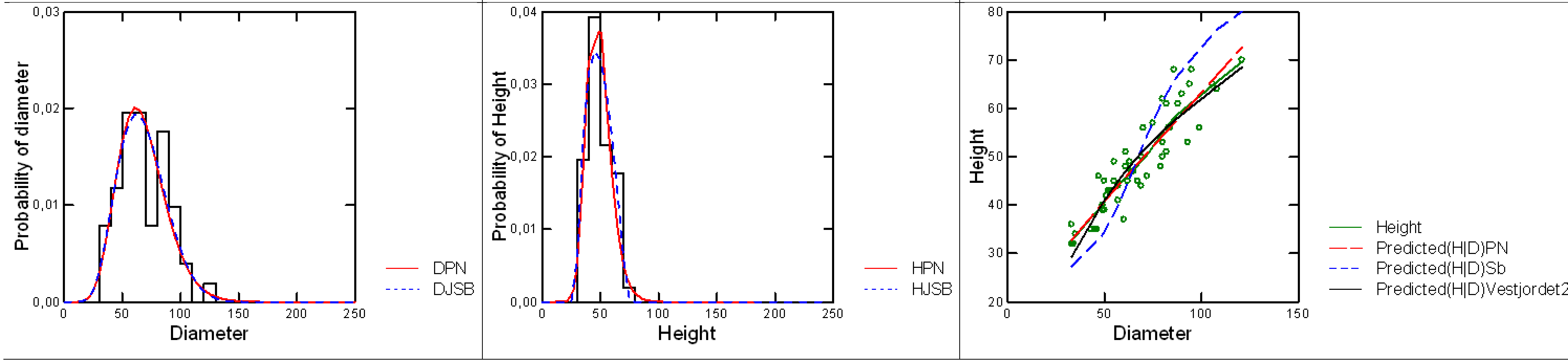




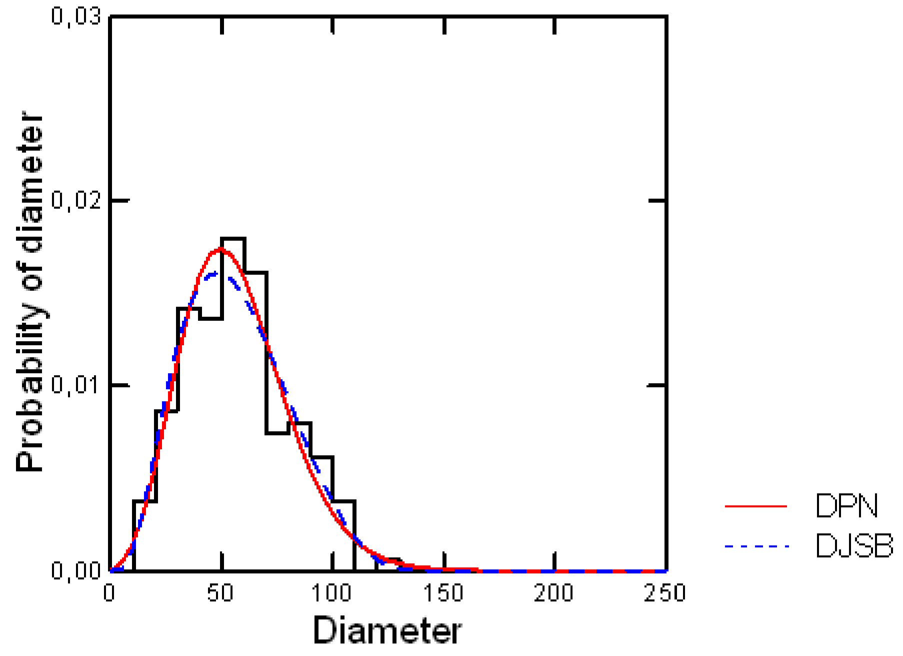
Results (↓) for STAND = 111

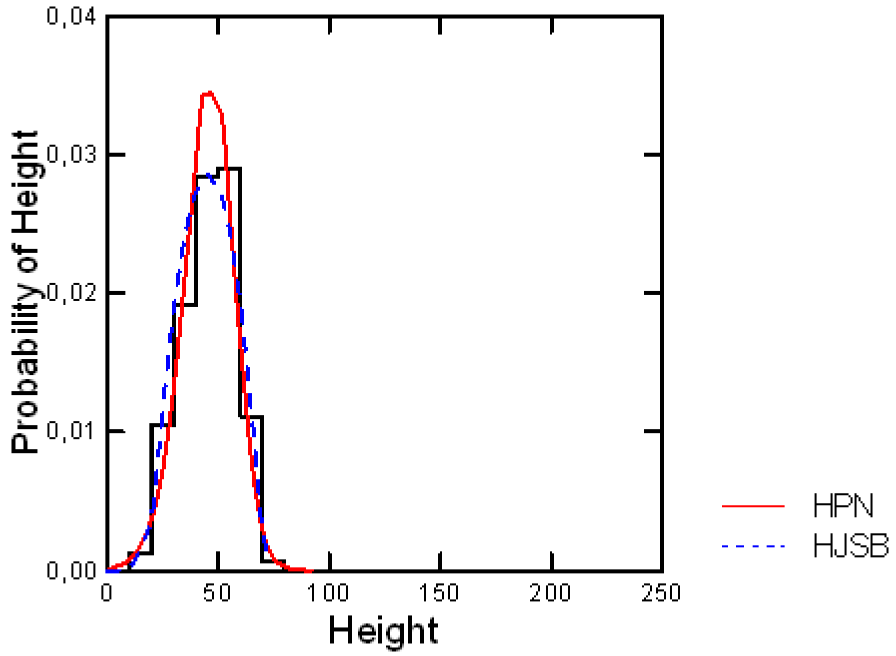


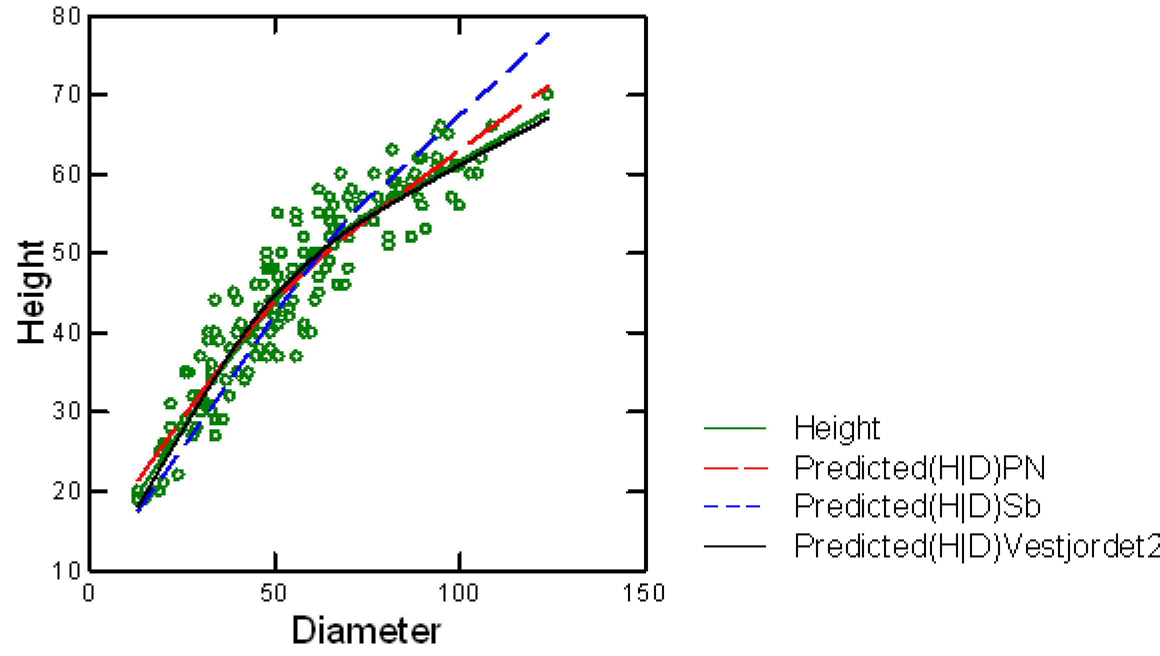
Results (↓) for STAND = 112



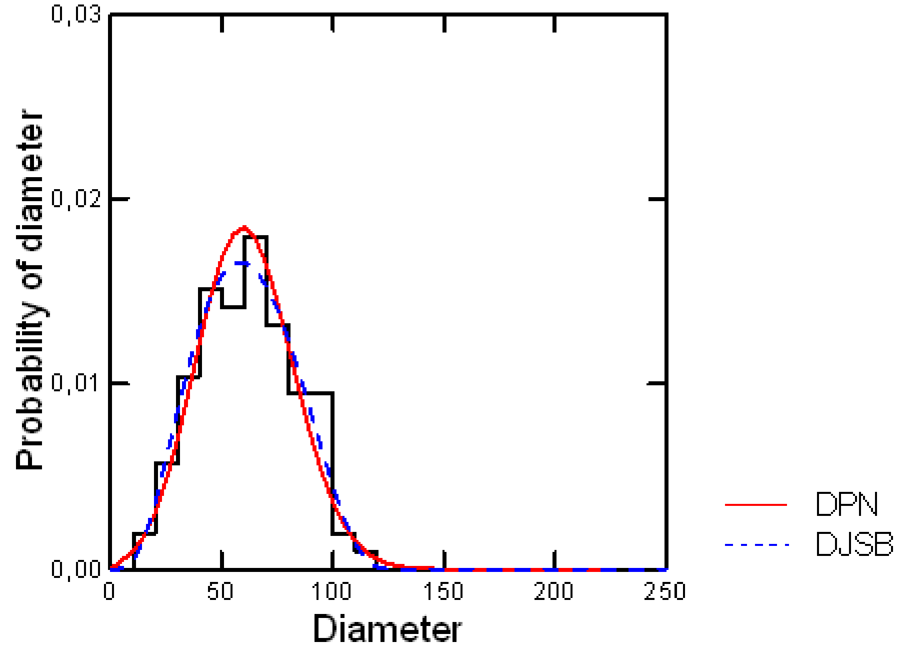
Results (↓) for STAND = 113

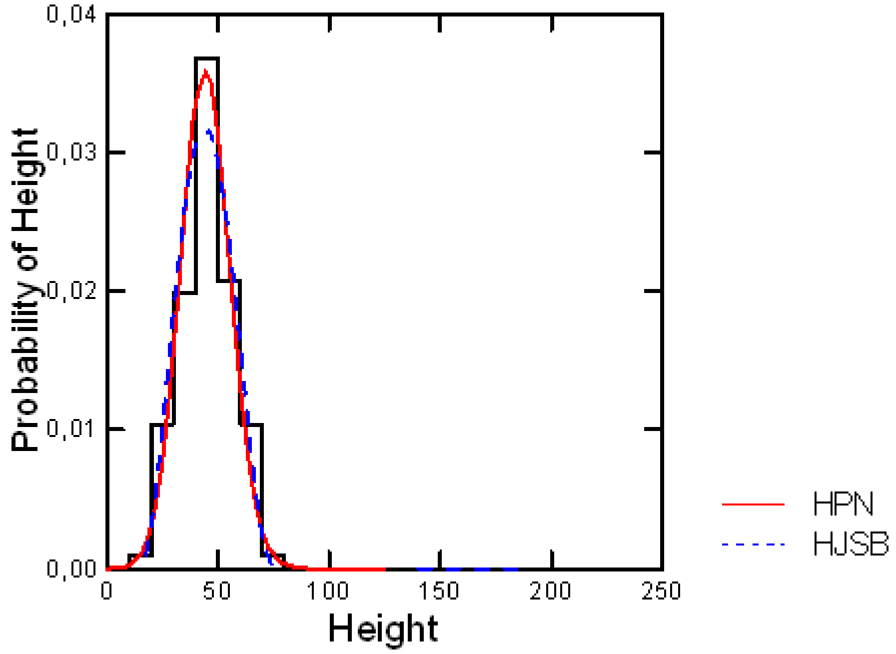


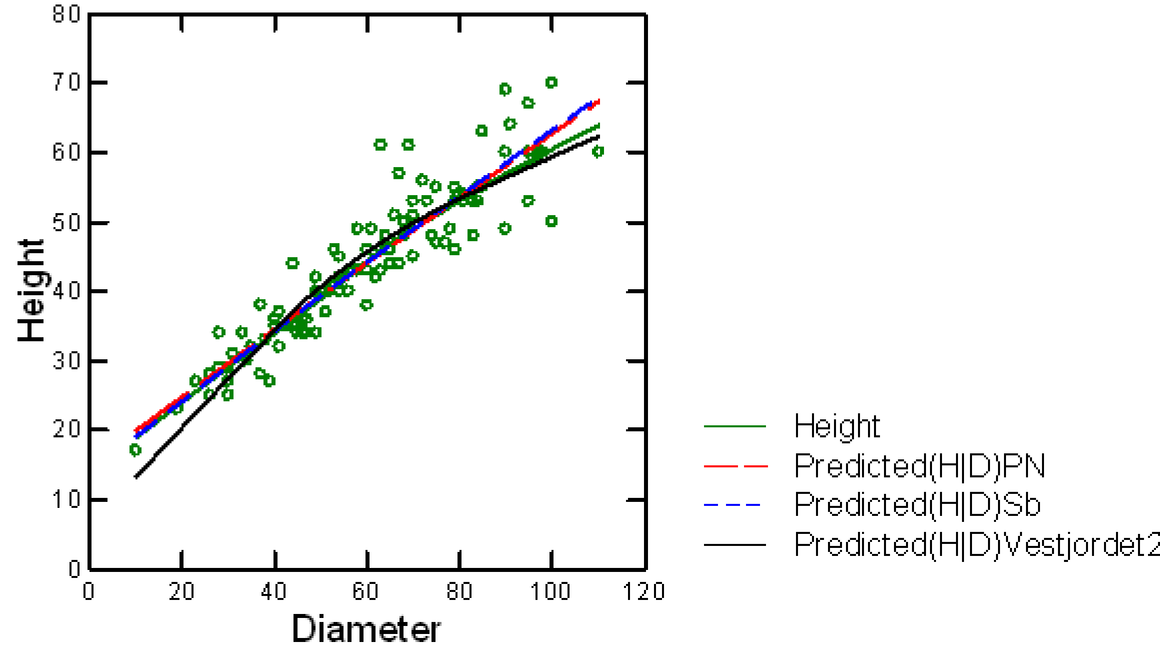




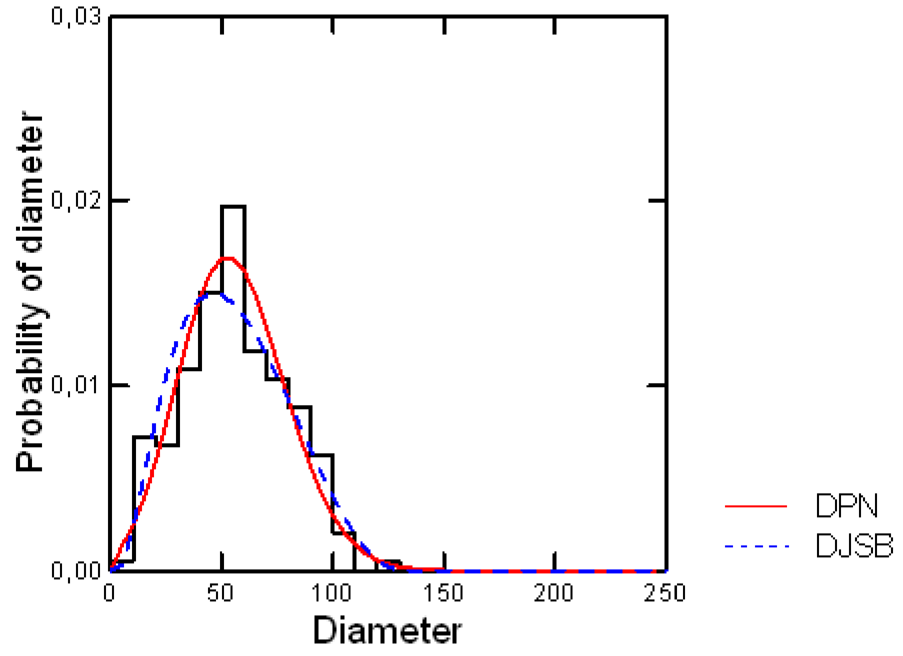
Results (↓) for STAND = 114

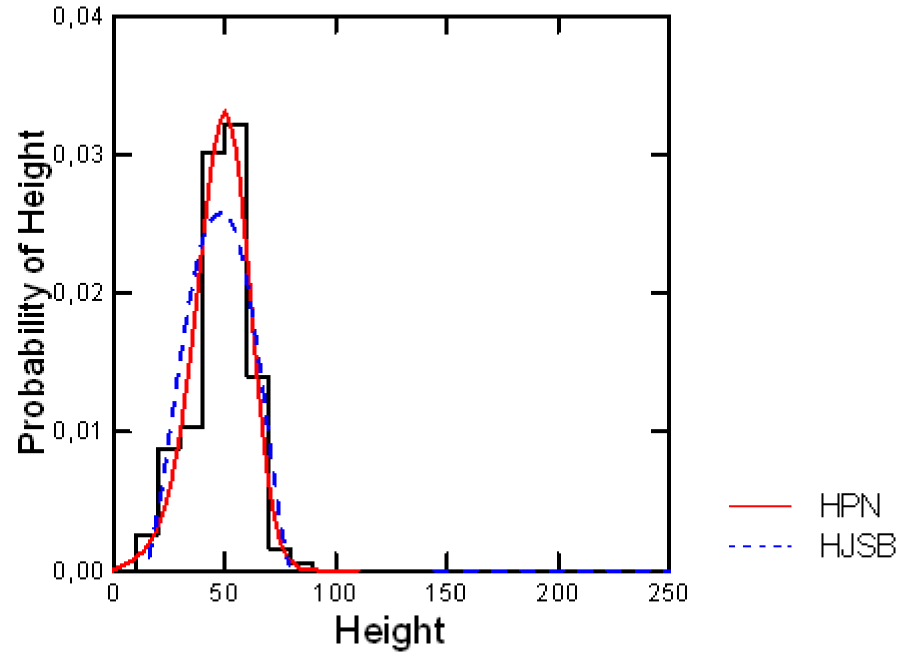


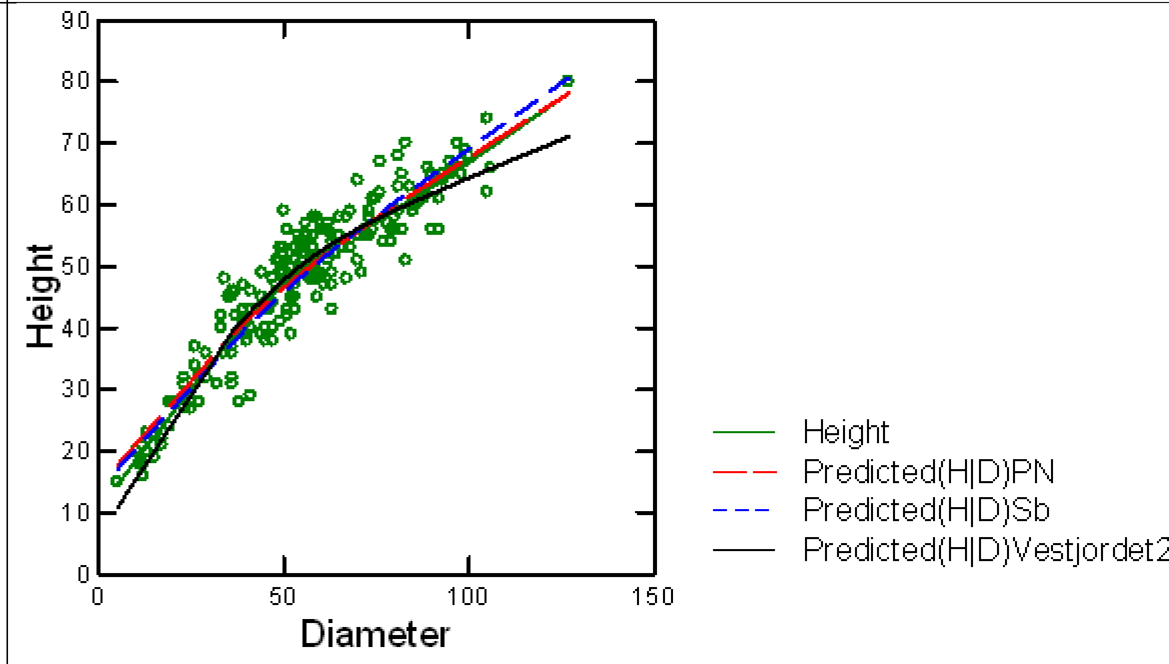




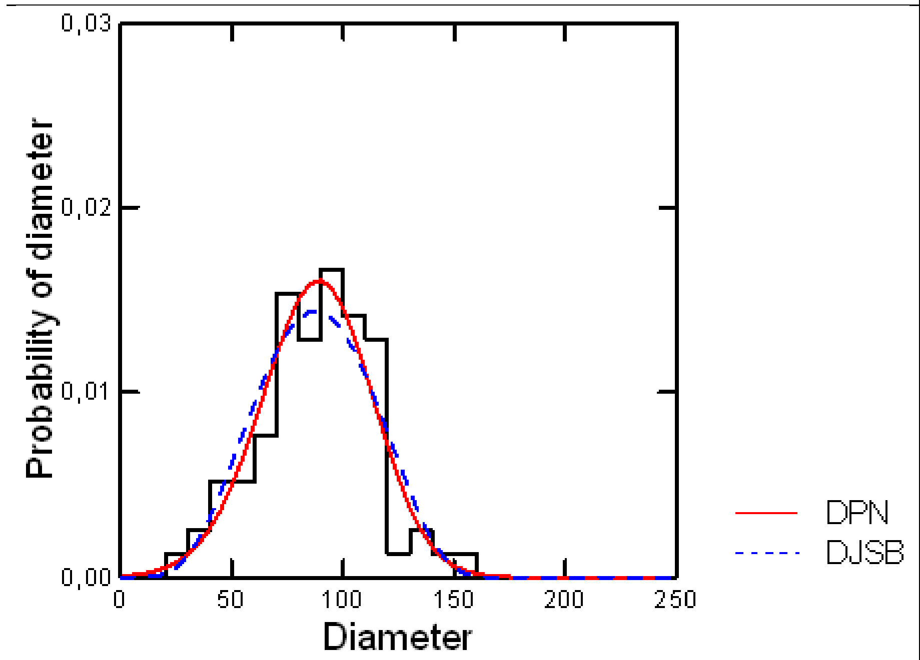
Results (↓) for STAND = 115

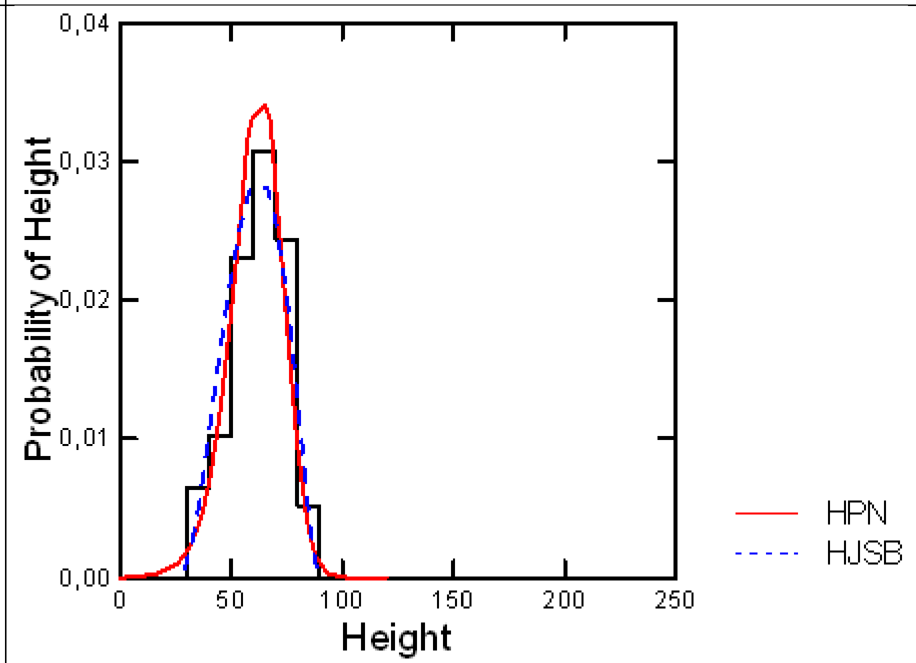


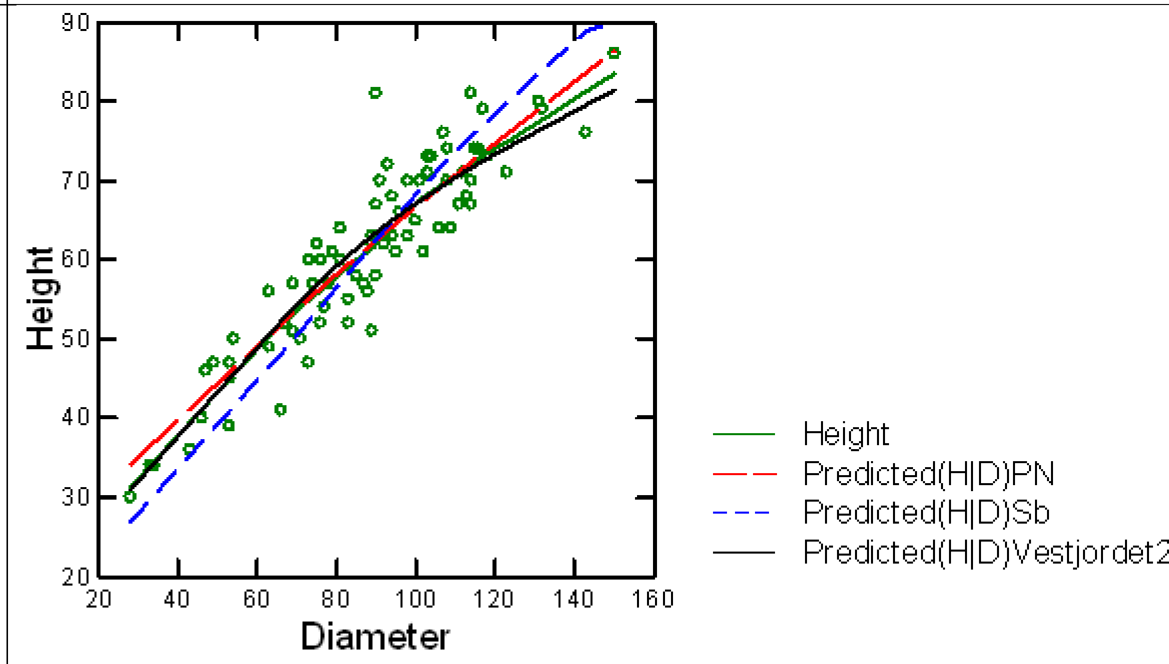




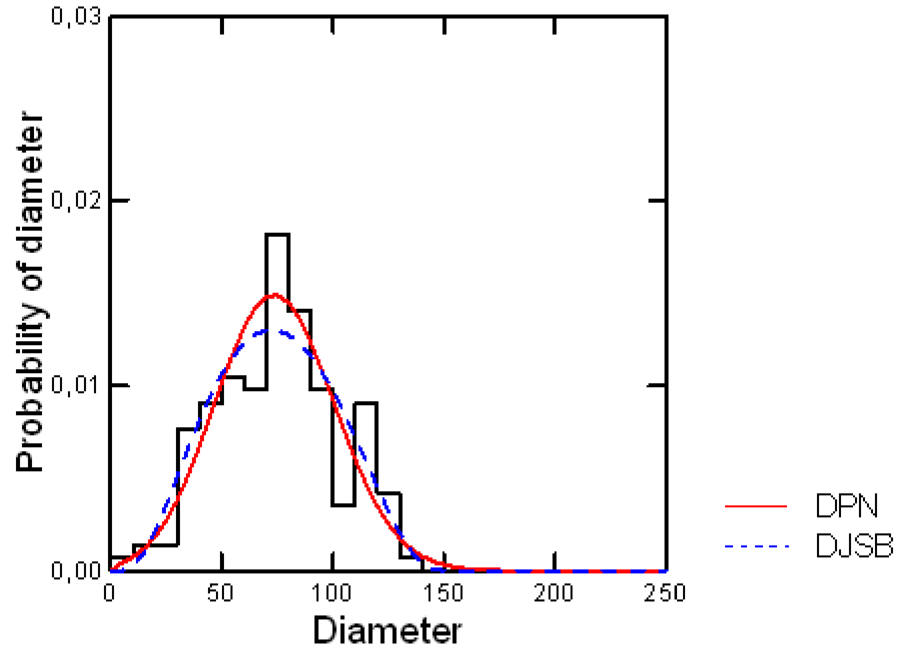
Results (↓) for STAND = 116

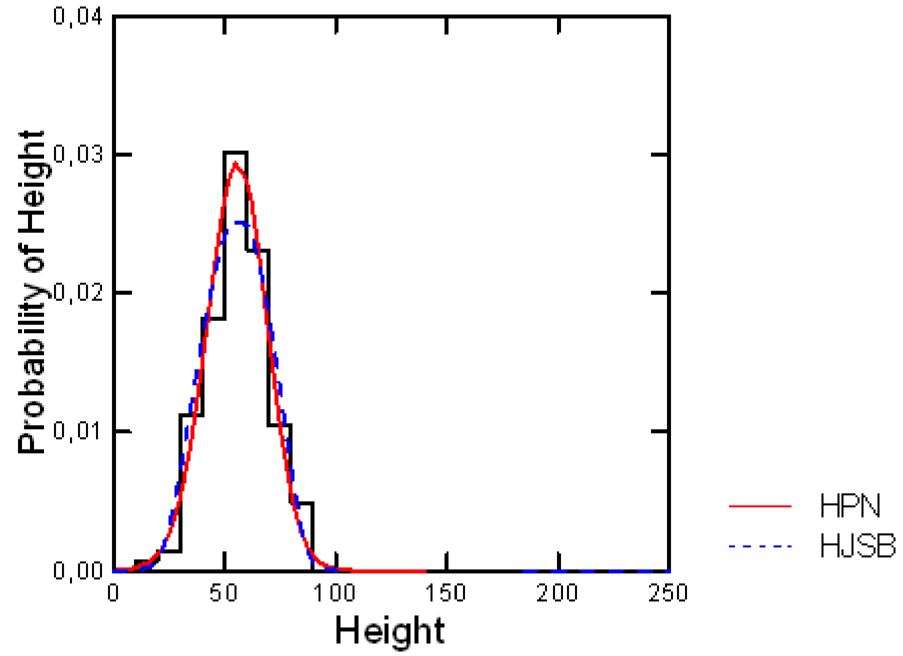


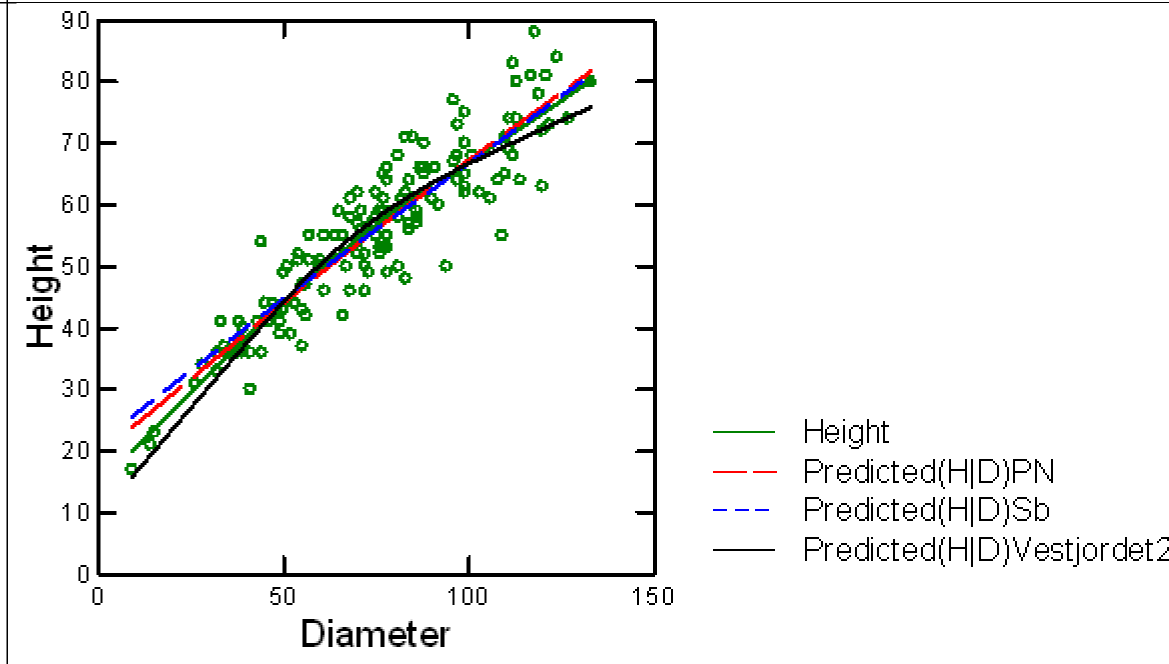




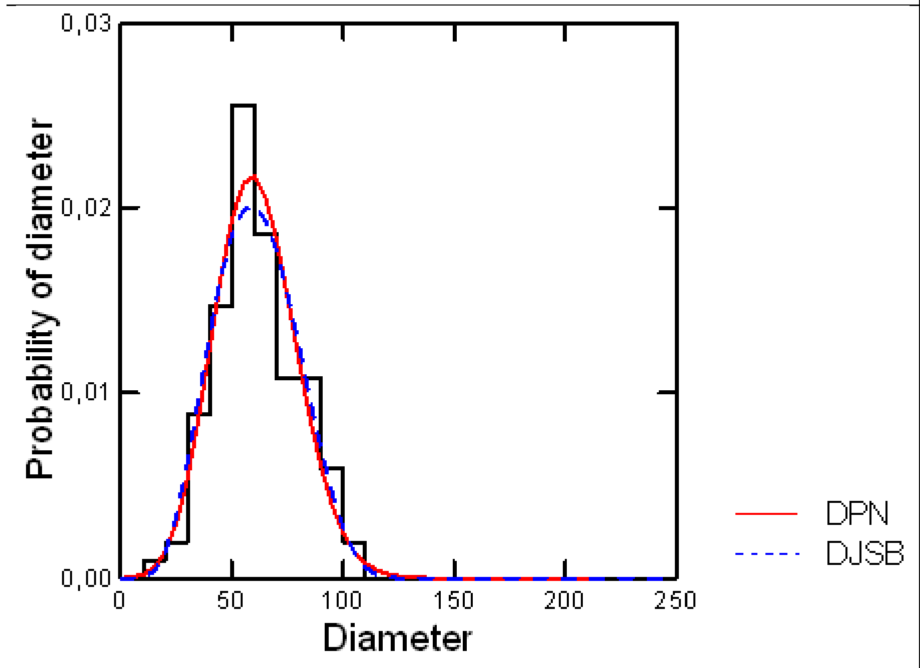
Results (↓) for STAND = 117

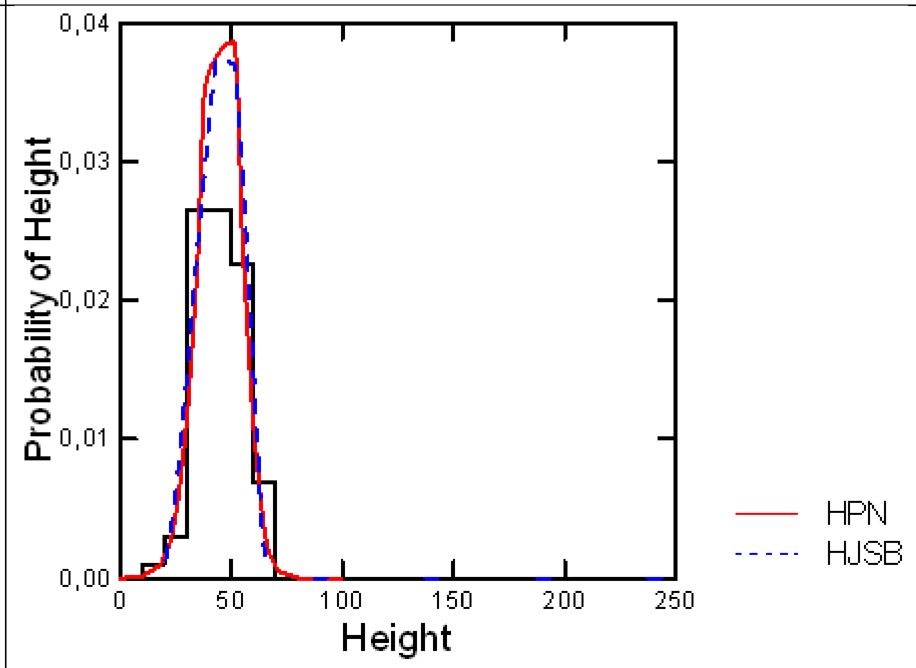


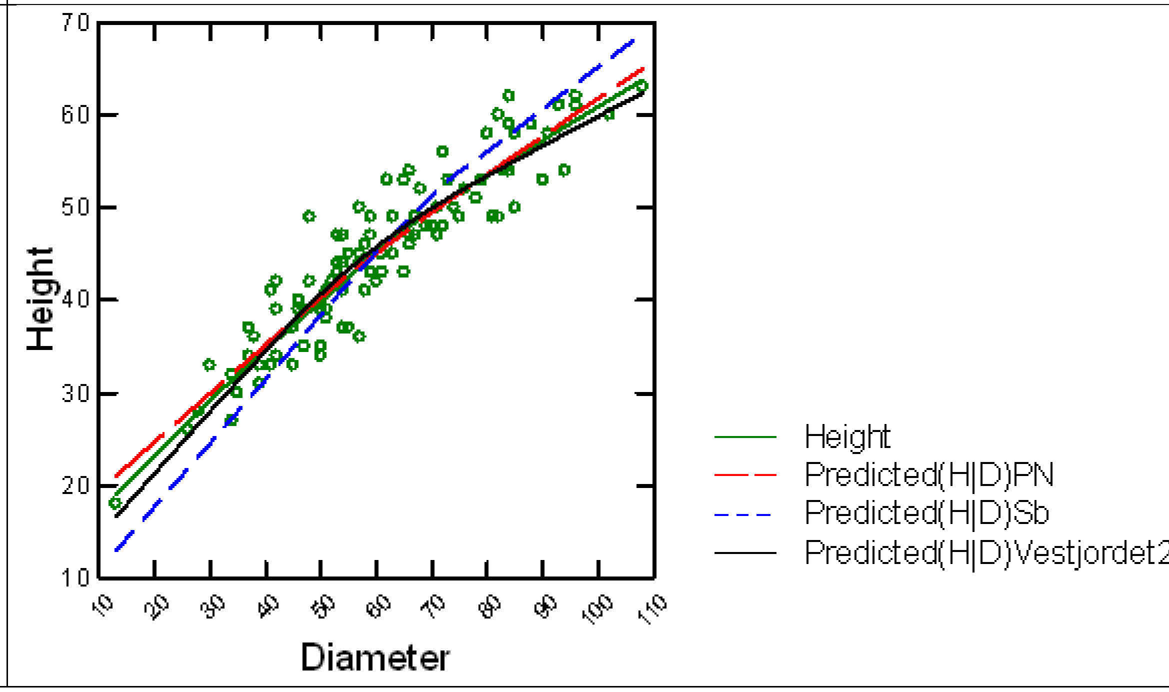




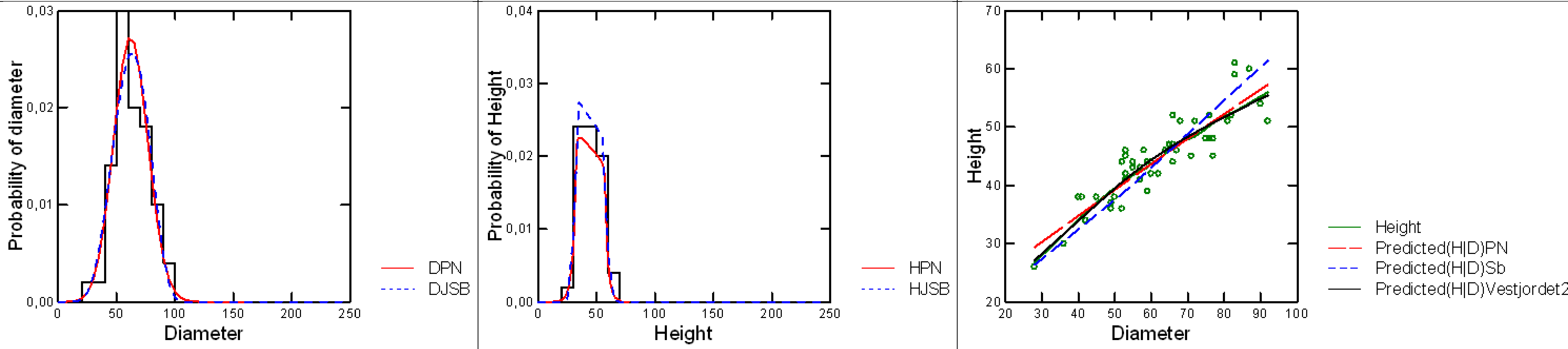
Results (↓) for STAND = 118



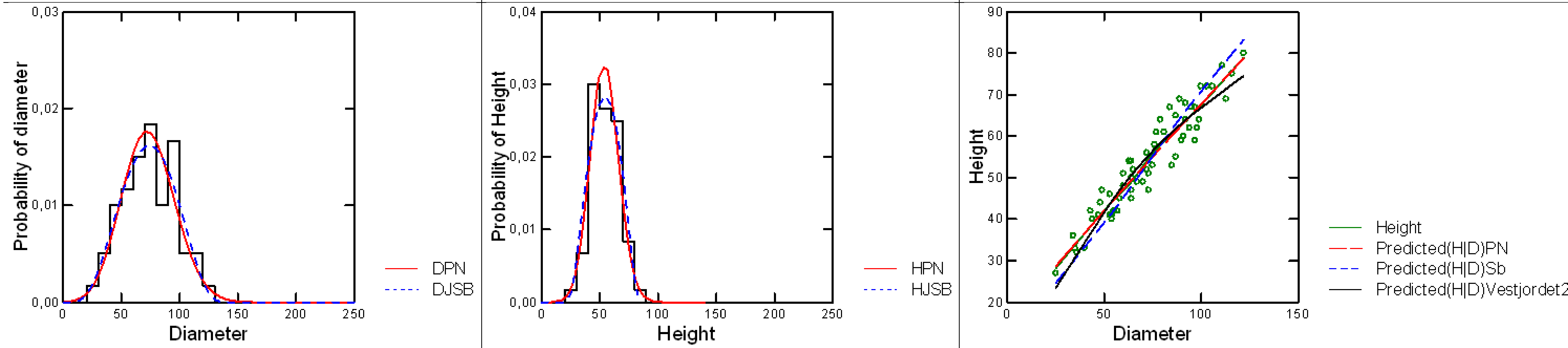




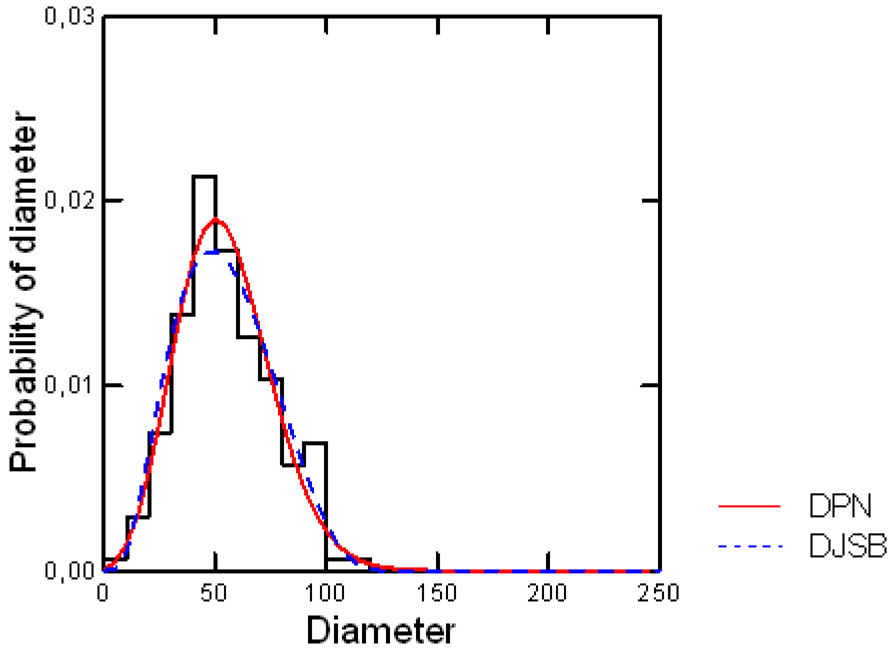
Results (↓) for STAND = 119

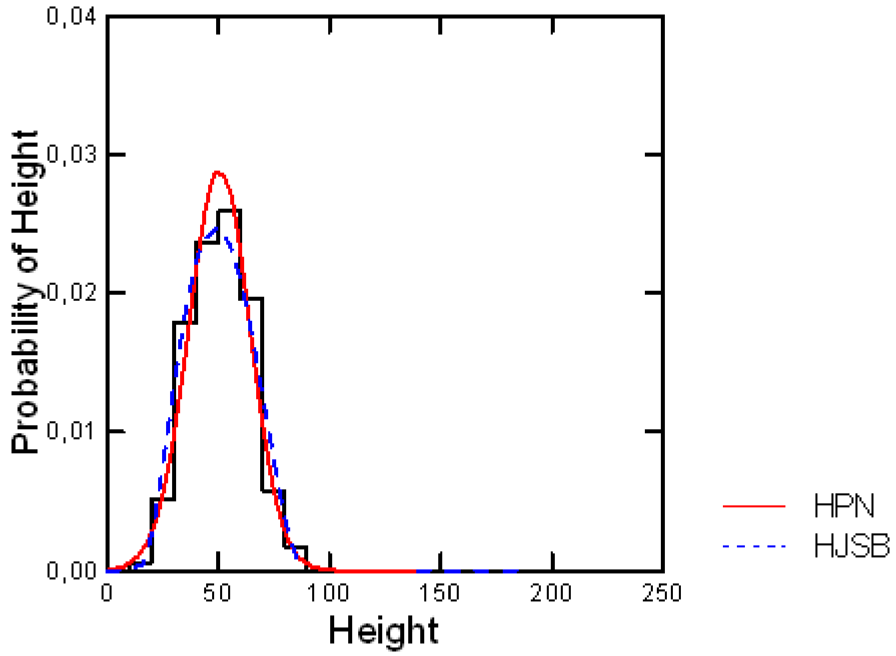


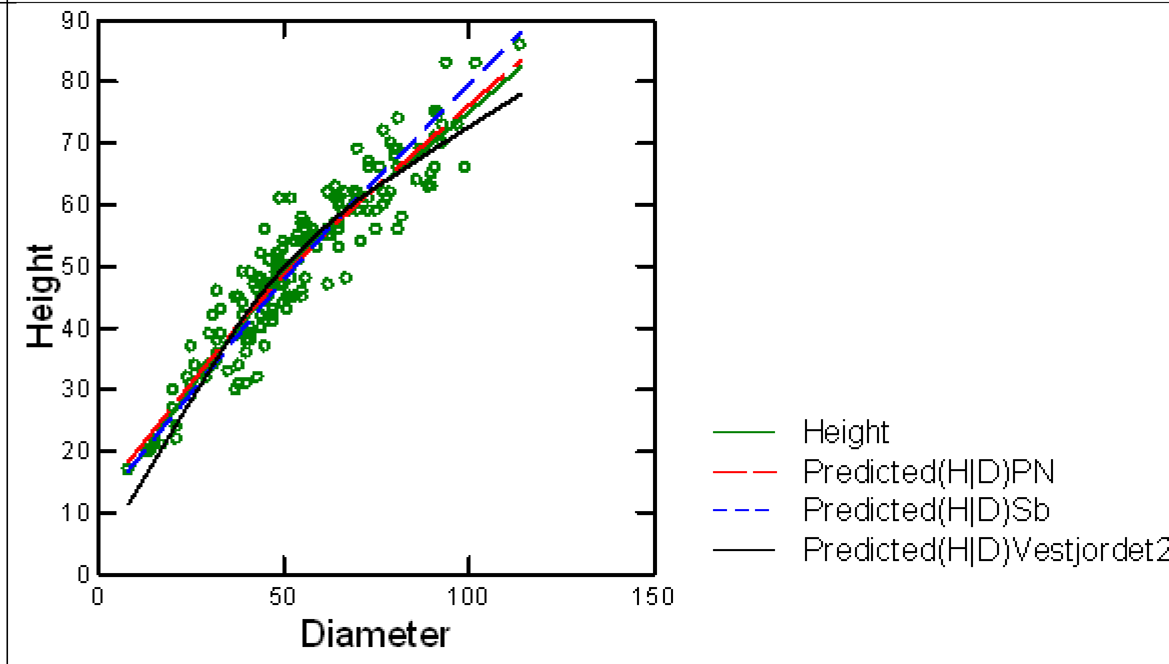
Results (↓) for STAND = 120



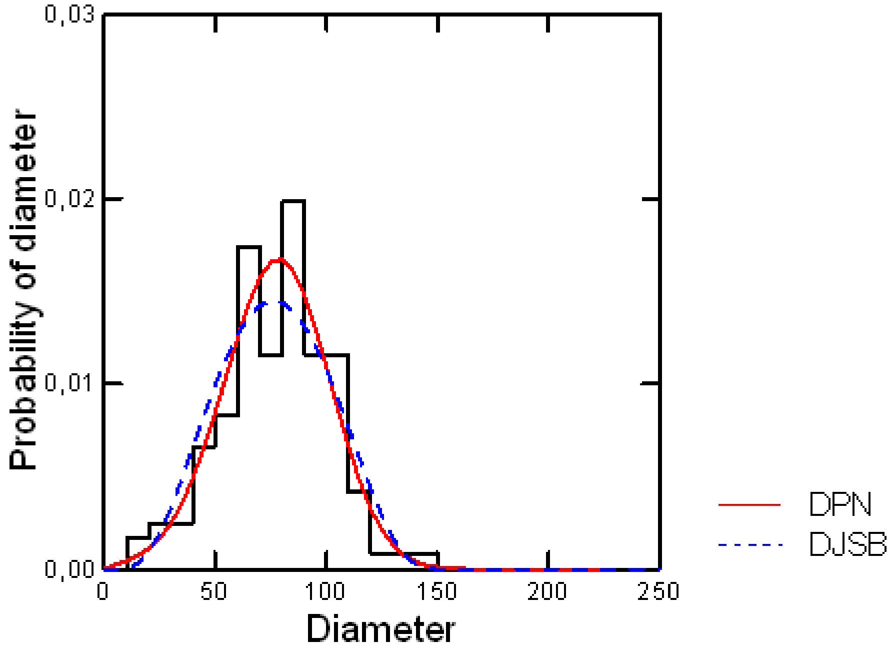
Results (↓) for STAND = 121

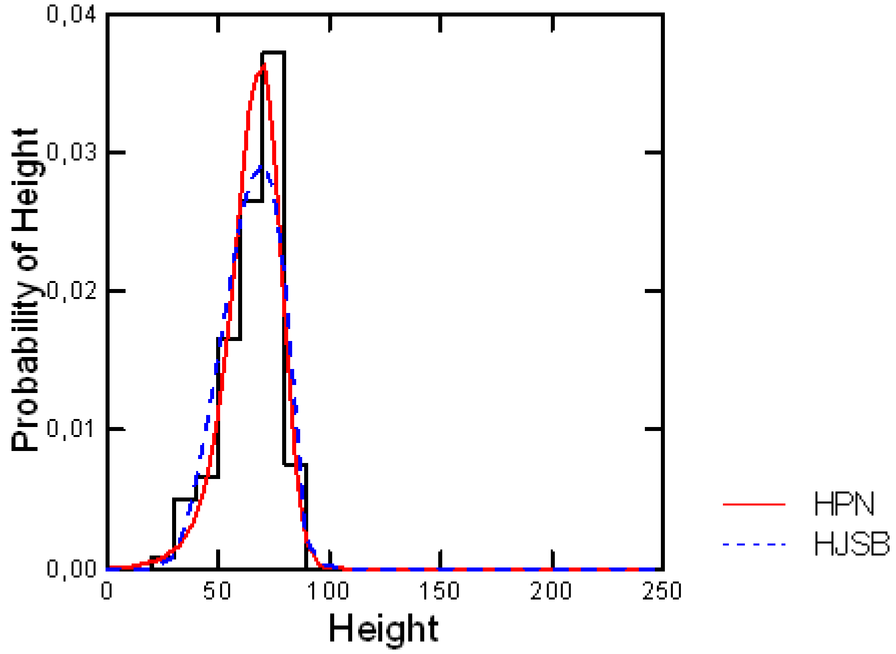


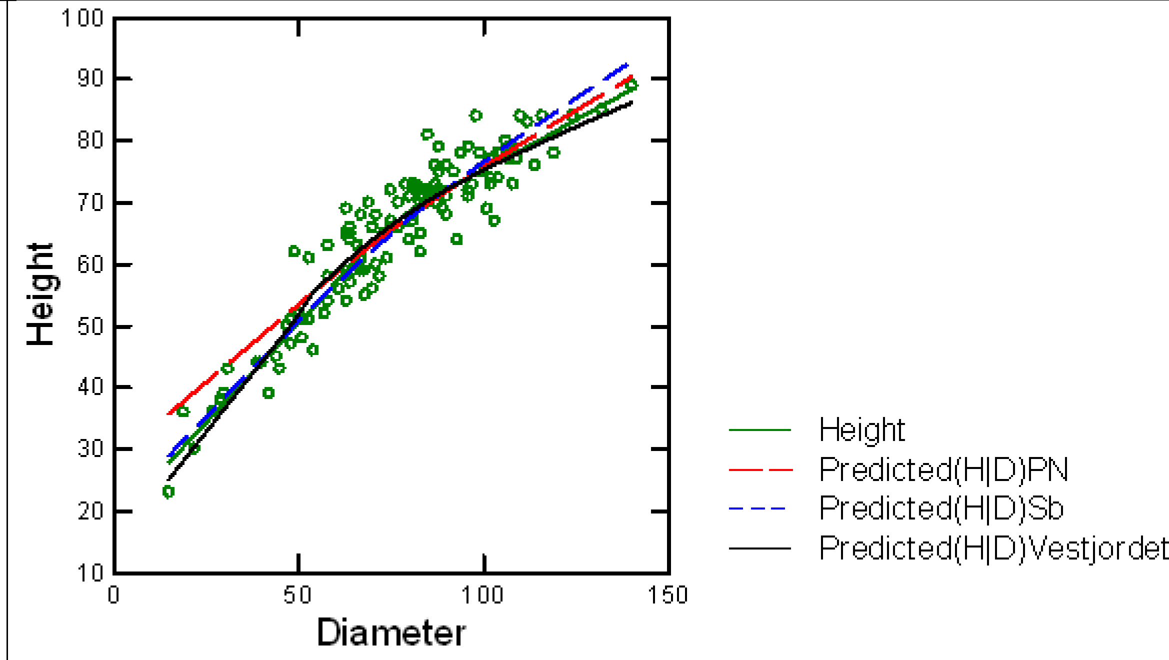




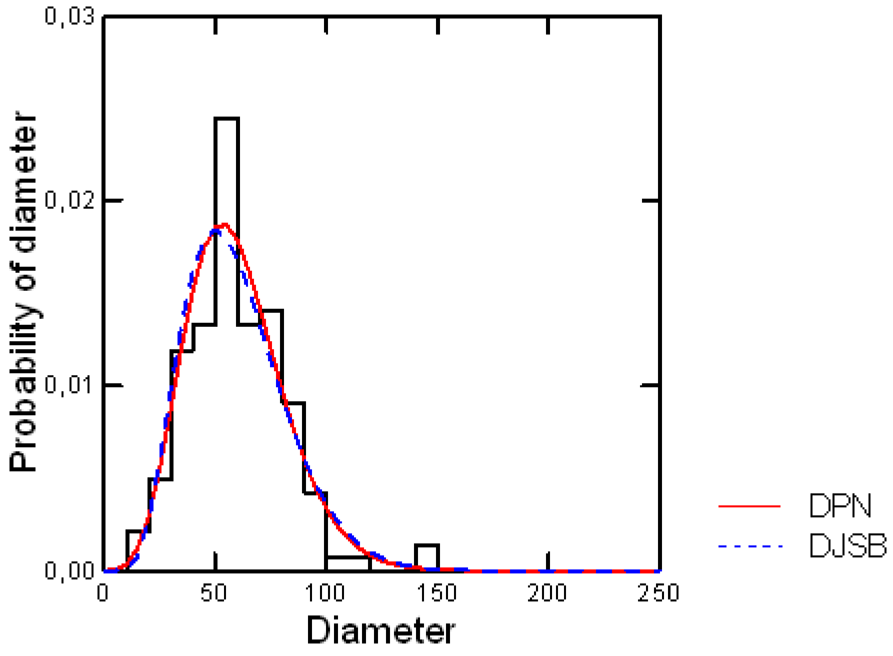
Results (↓) for STAND = 122

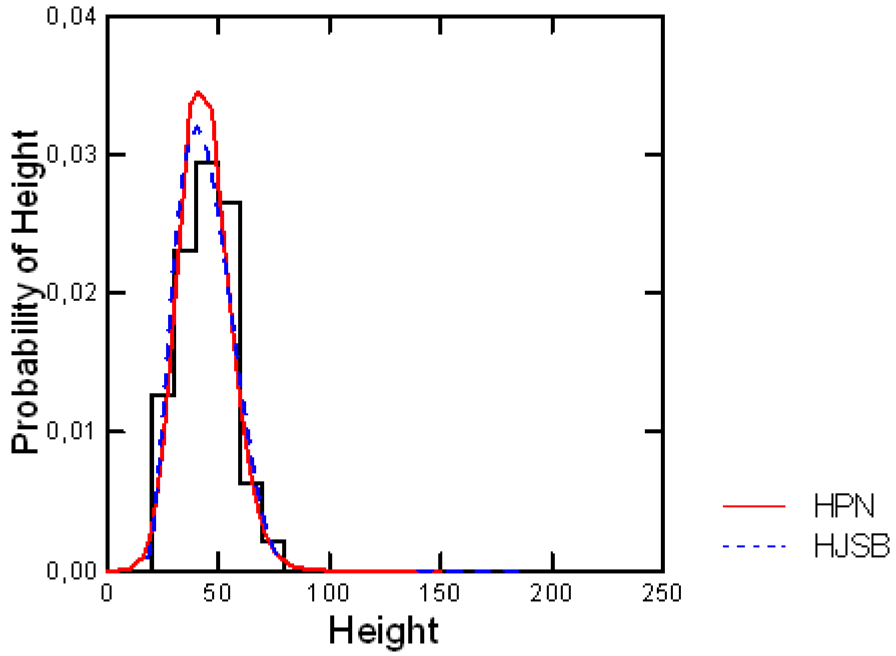


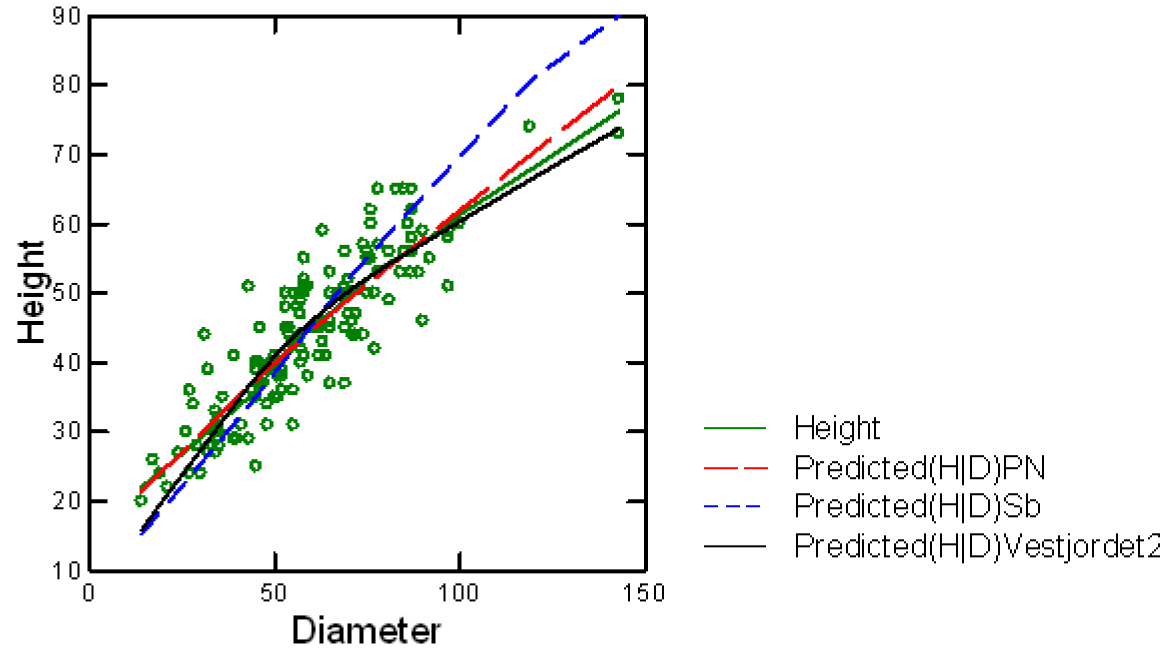




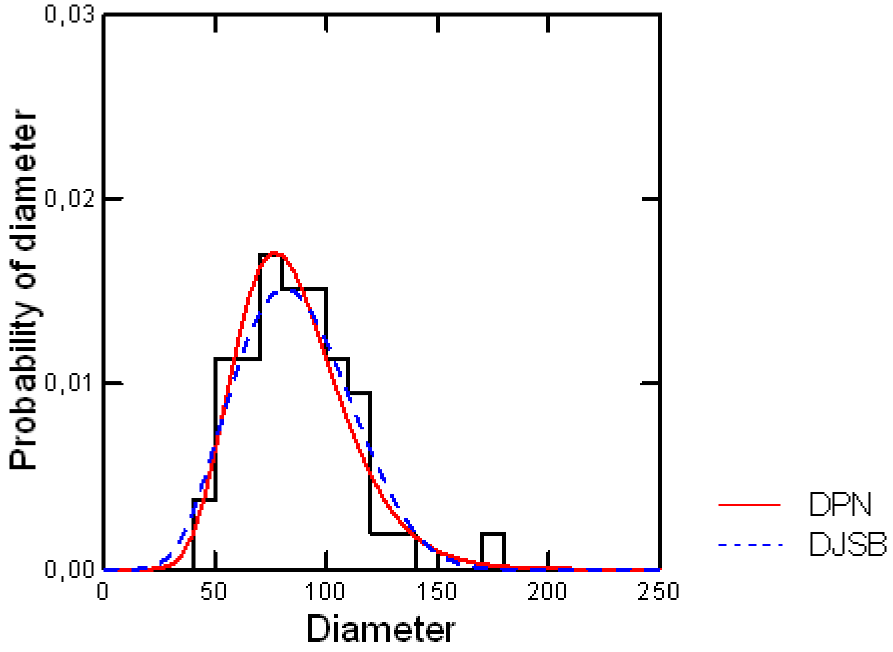
Results (↓) for STAND = 123

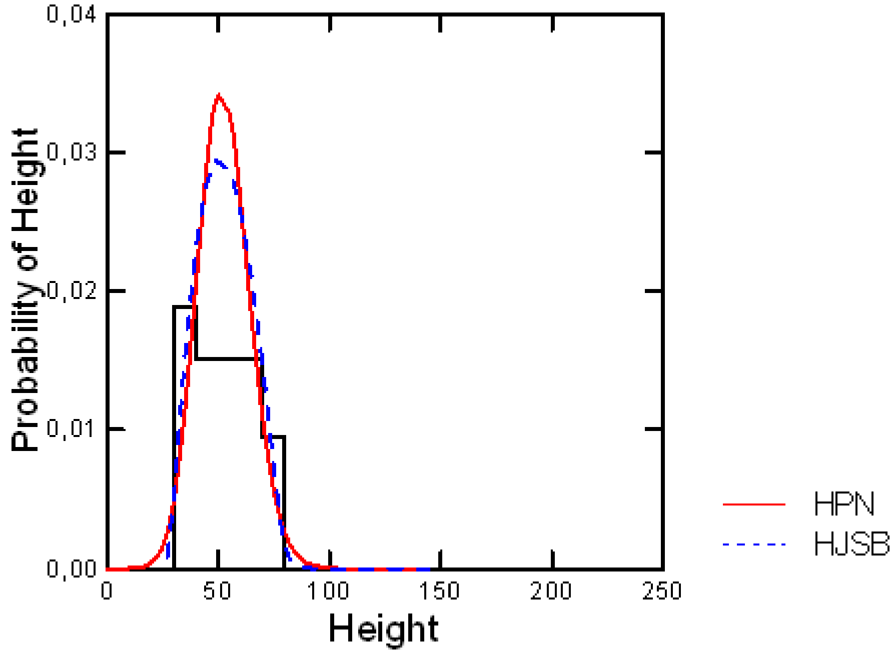


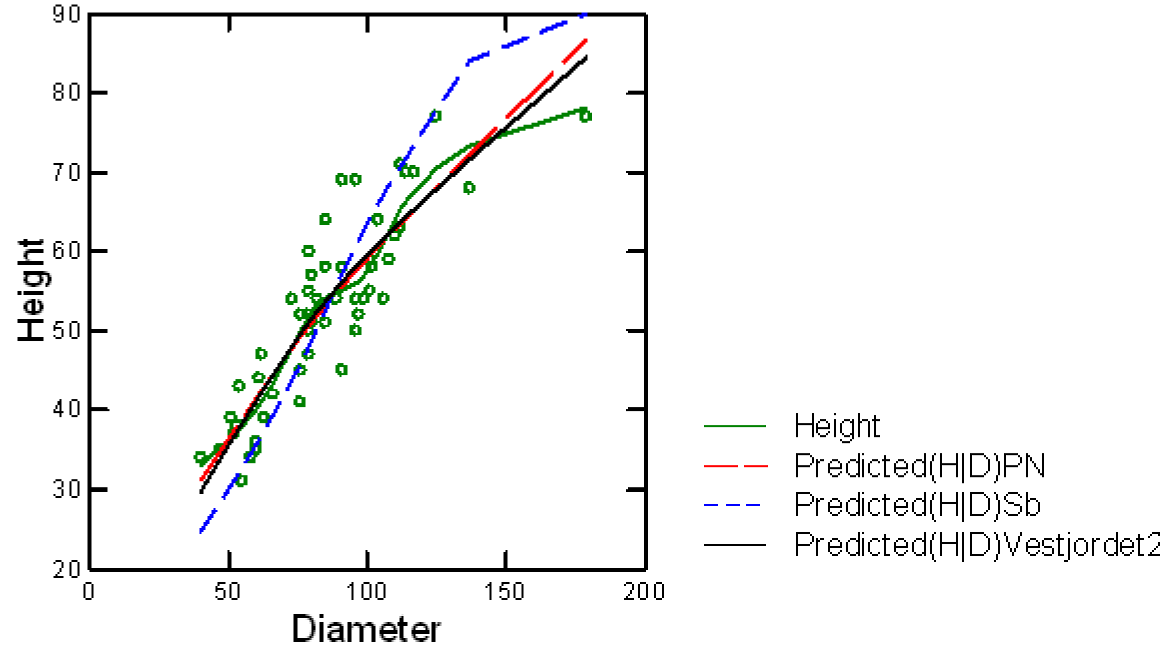




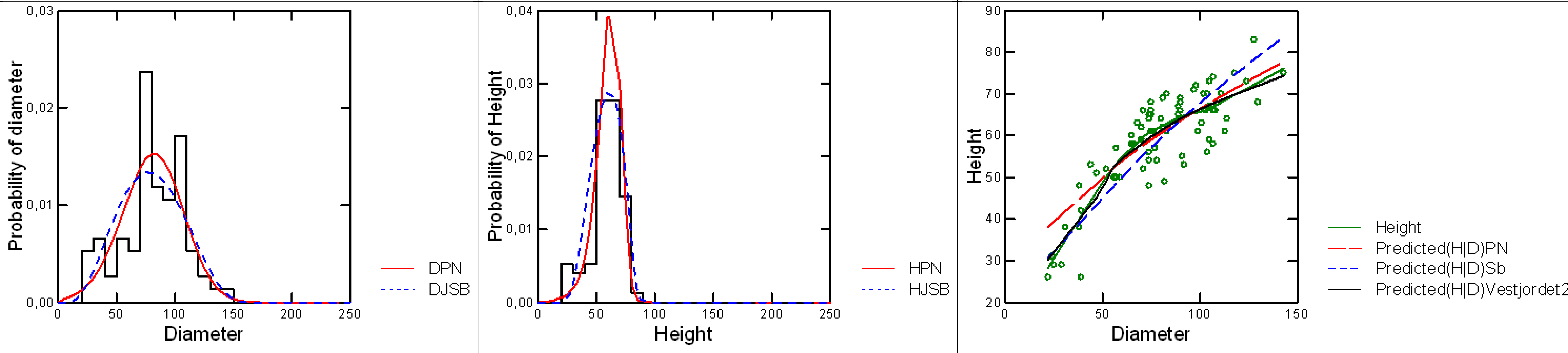
Results (↓) for STAND = 124



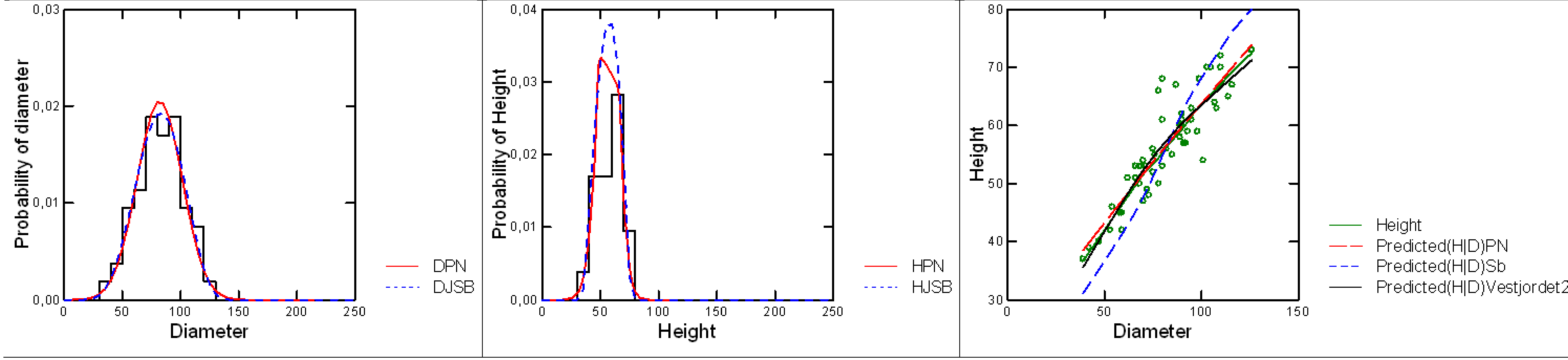




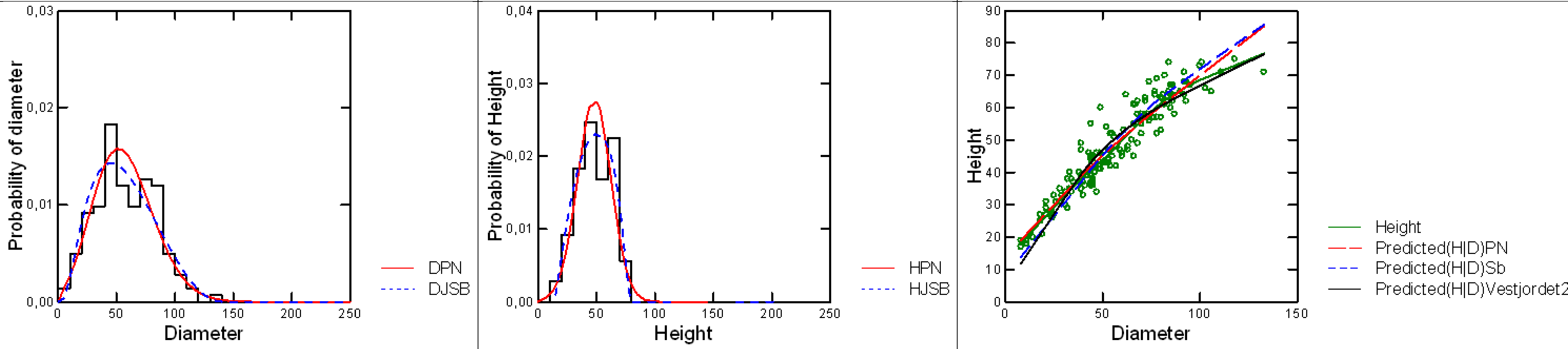
Results (↓) for STAND = 125



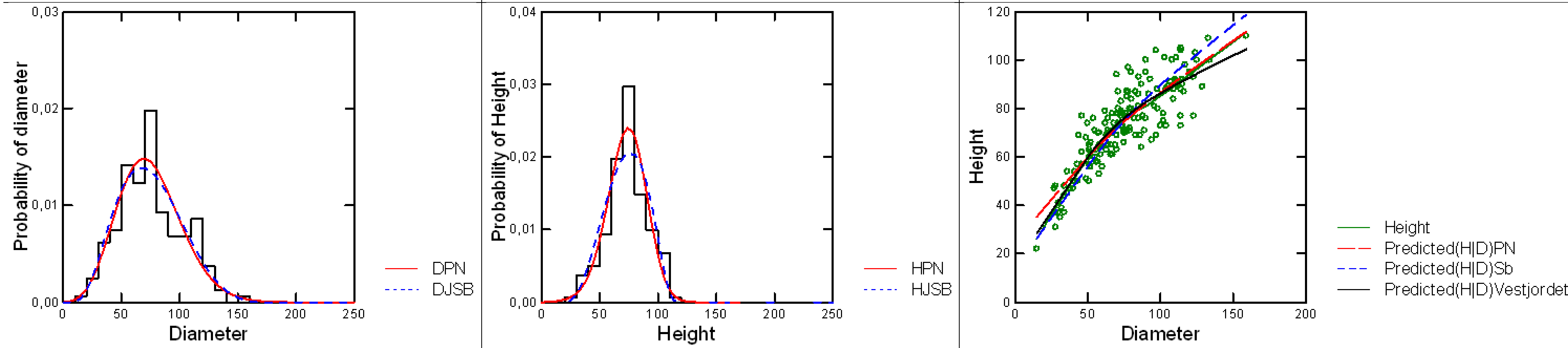
Results (↓) for STAND = 126



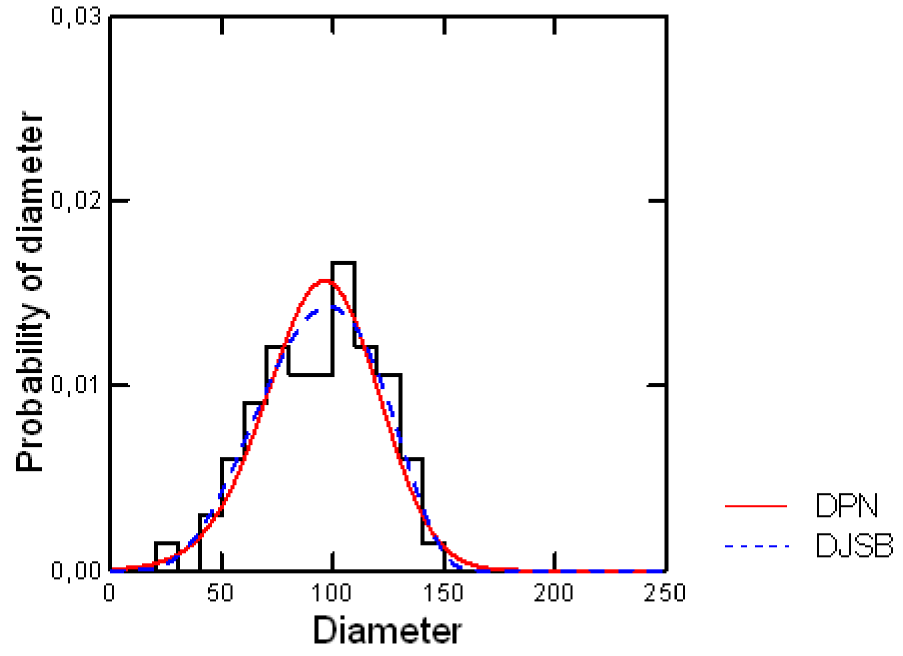
Results (↓) for STAND = 127

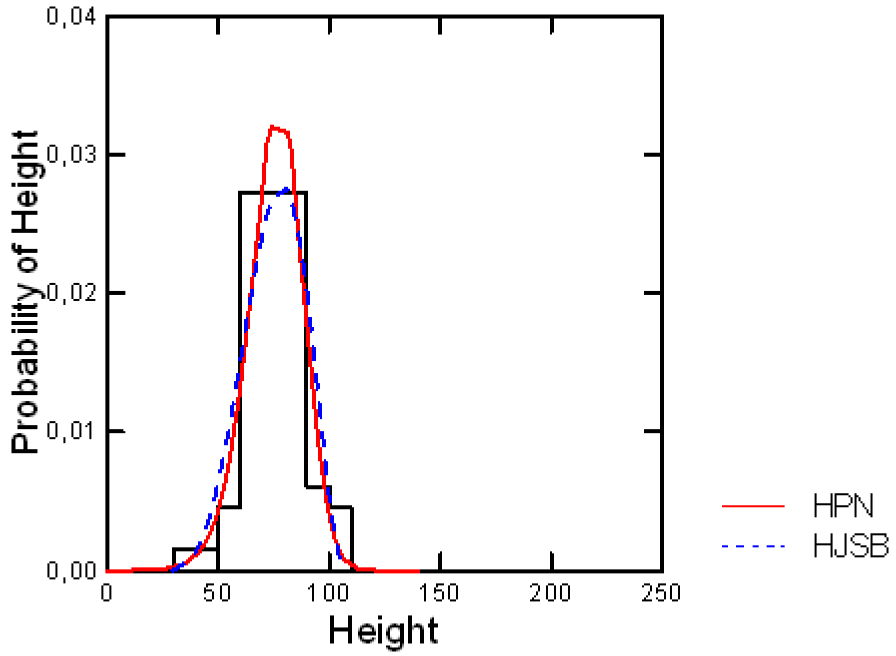


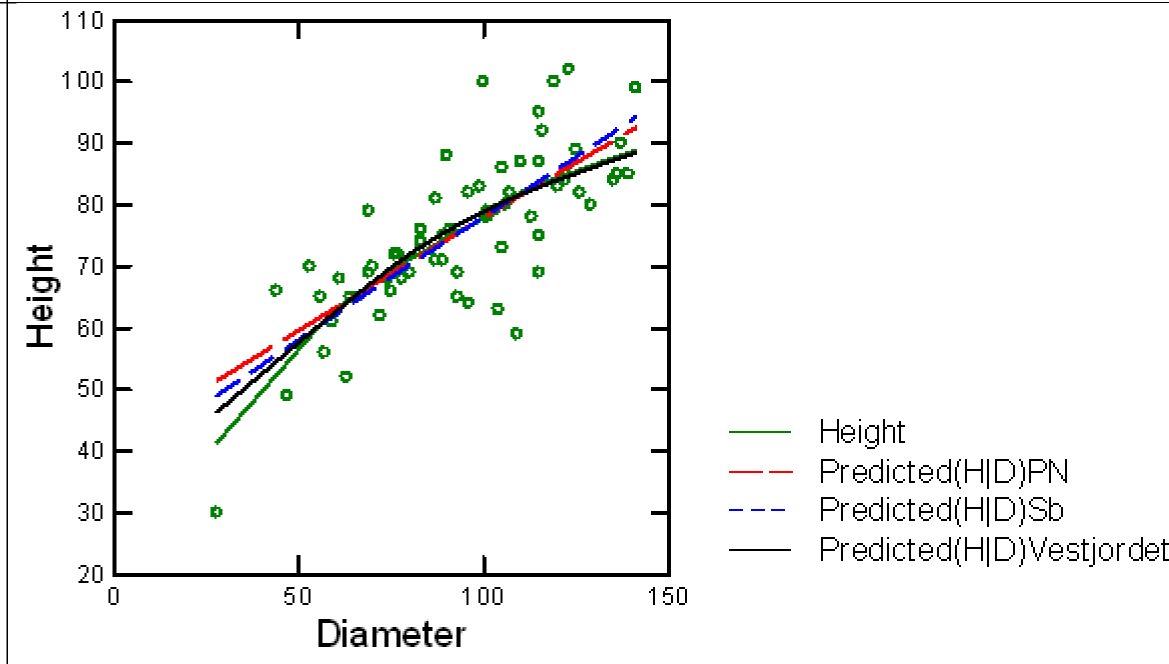
Results (↓) for STAND = 128



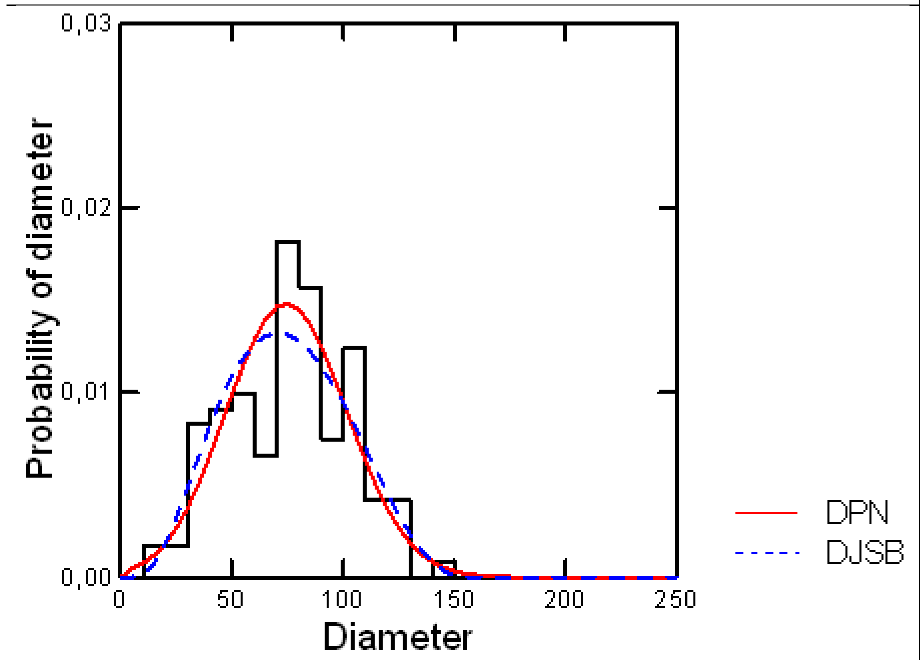
Results (↓) for STAND = 129

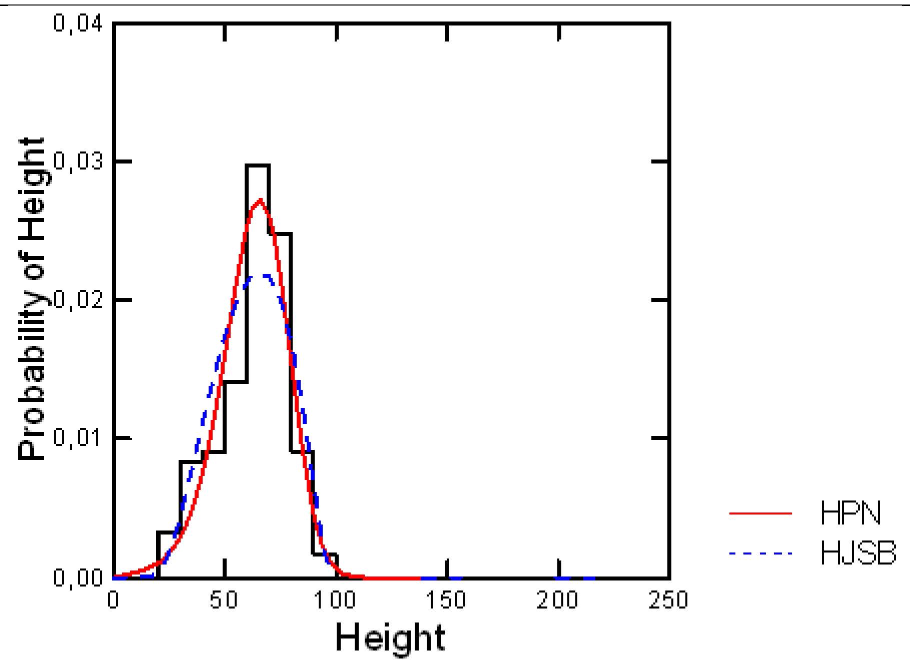


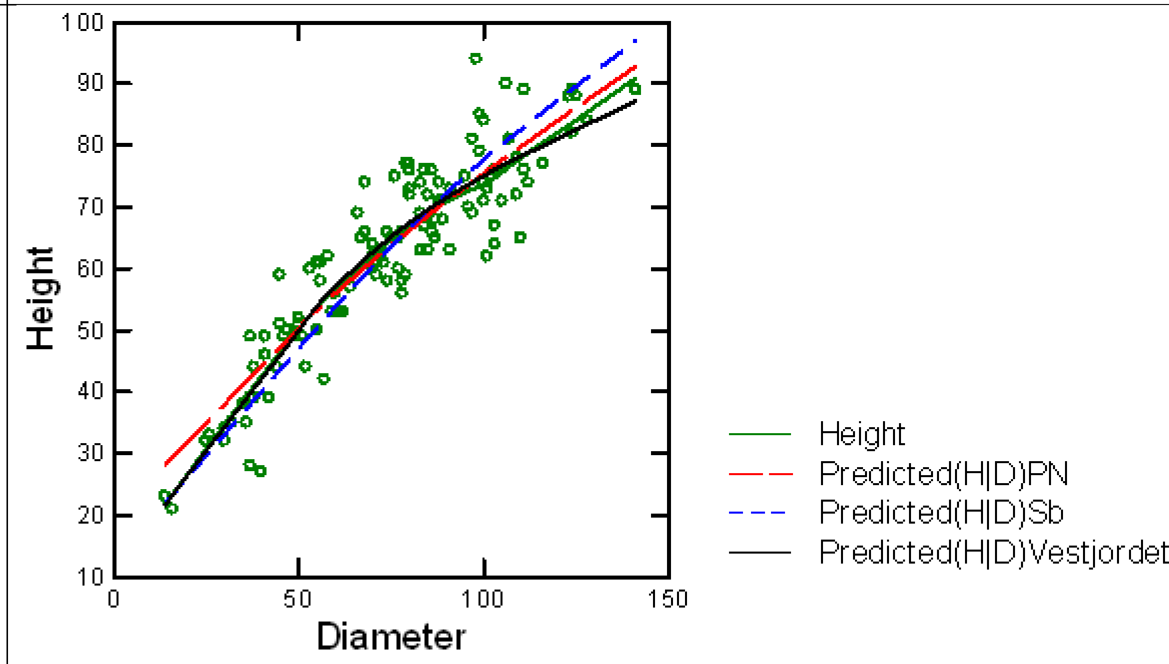




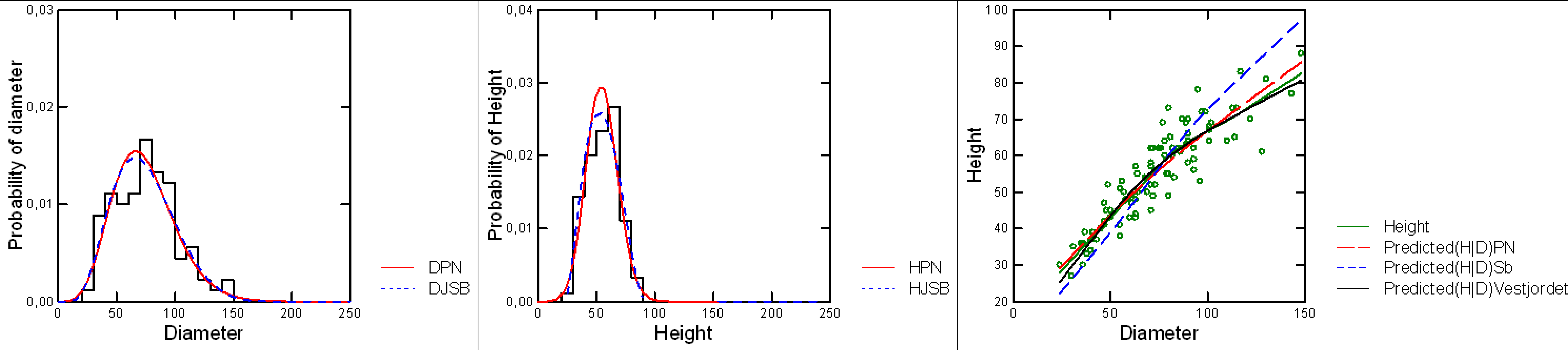
Results (↓) for STAND = 130



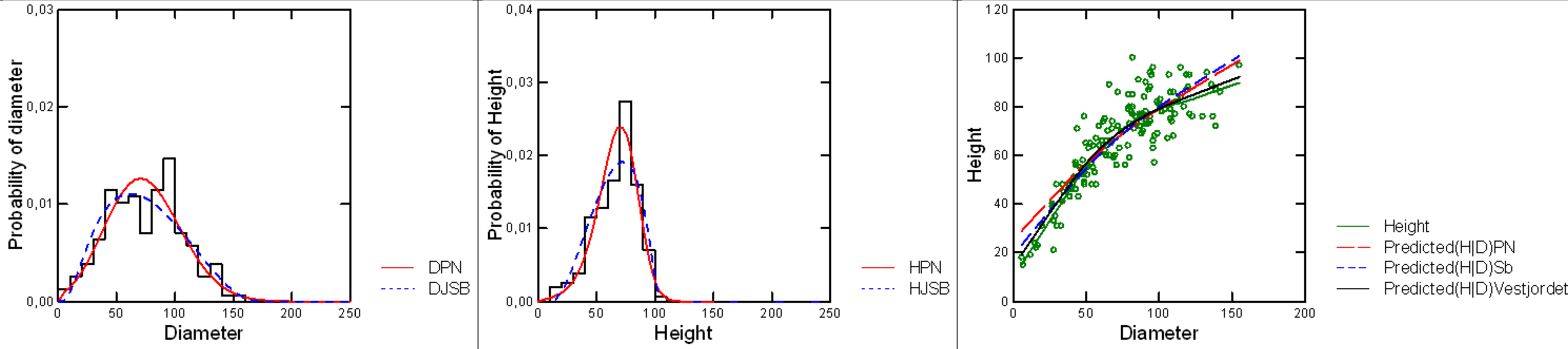




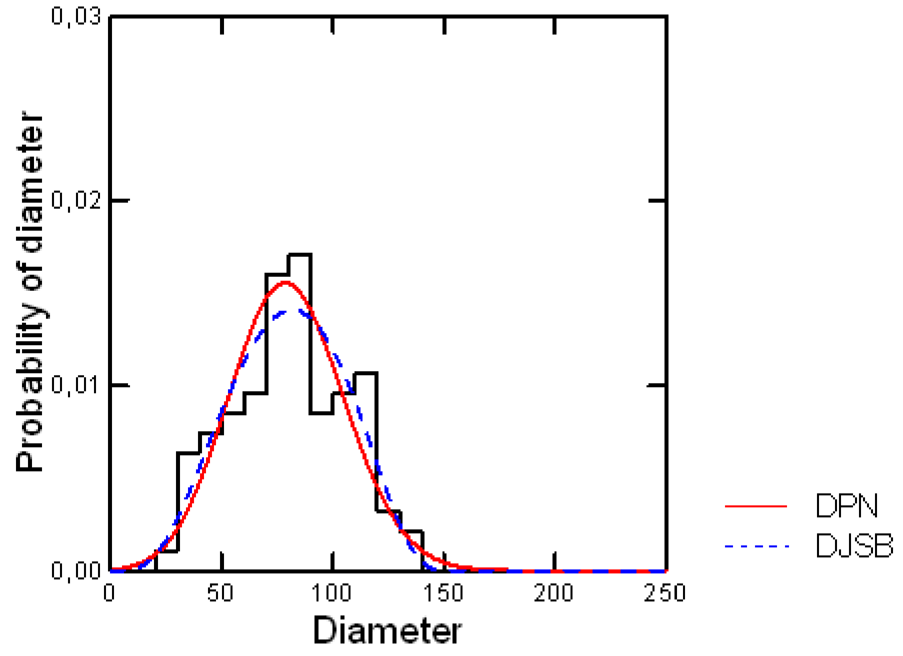
Results (↓) for STAND = 131

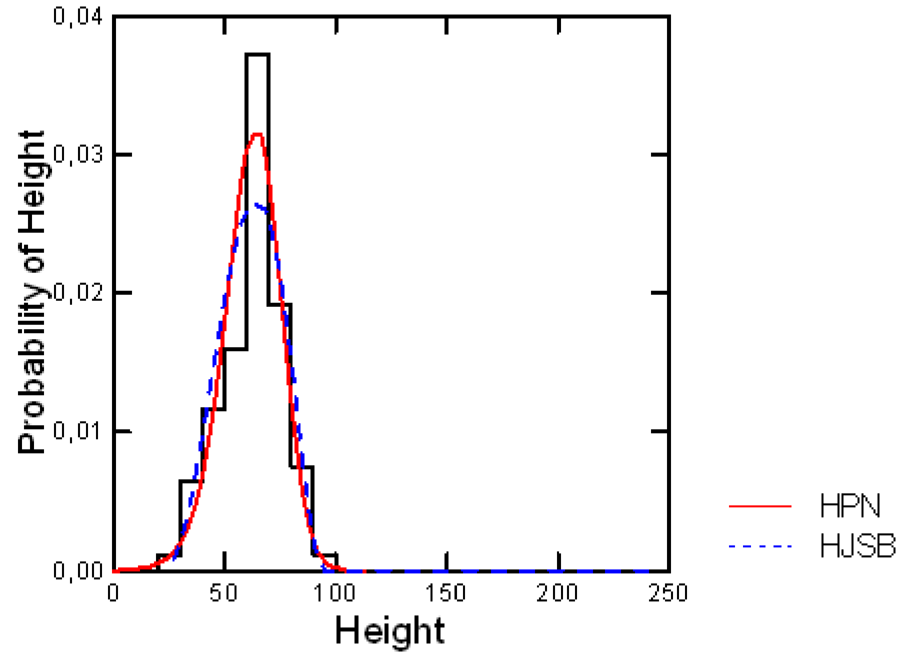


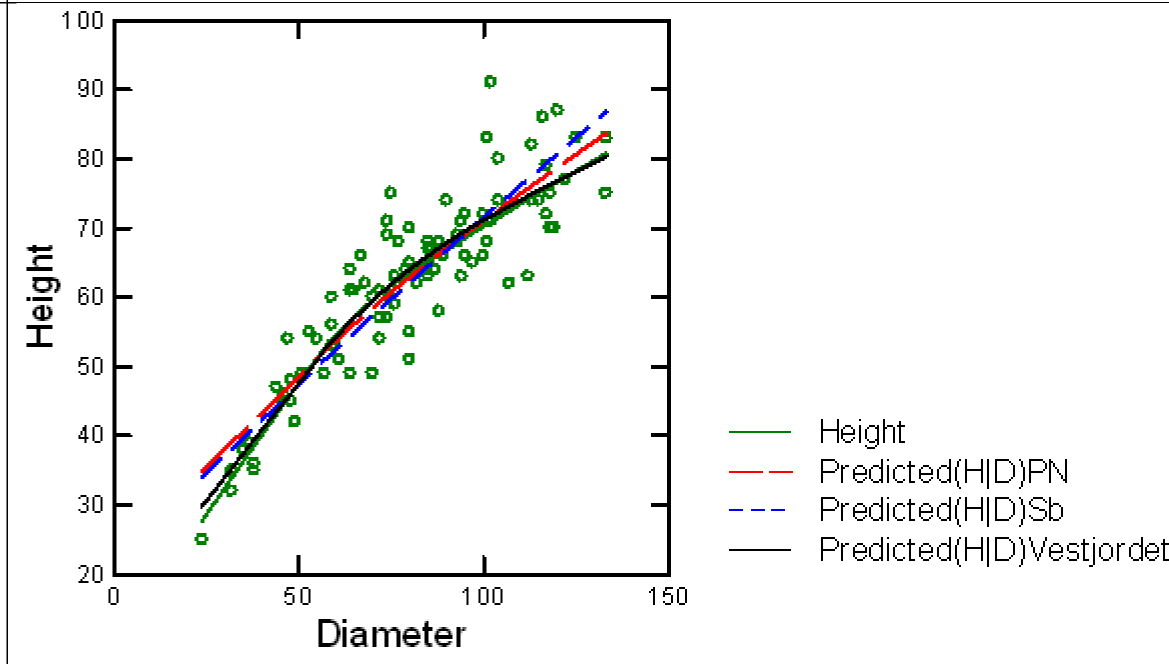
Results (↓) for STAND = 132



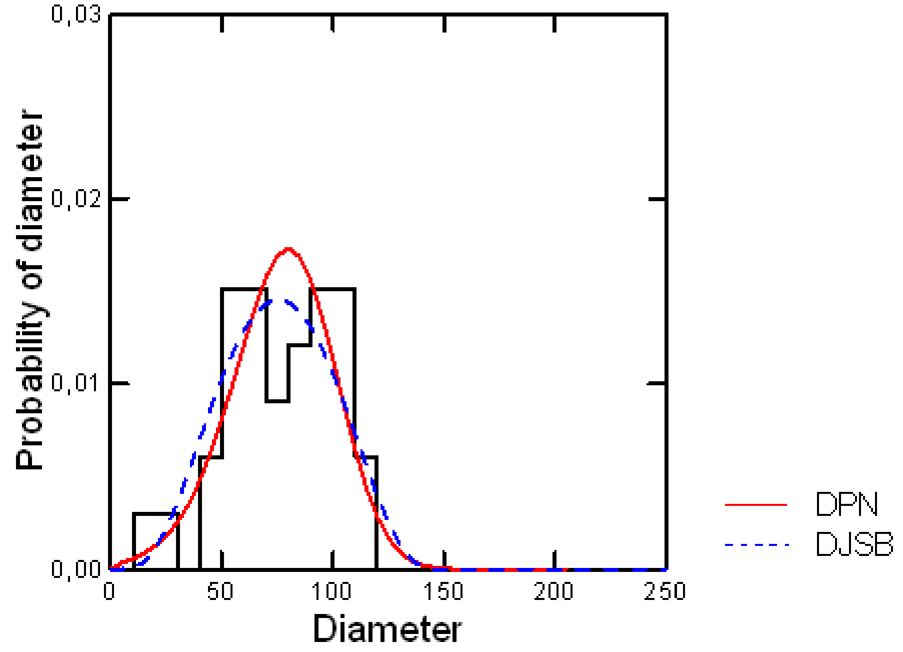
Results (↓) for STAND = 133

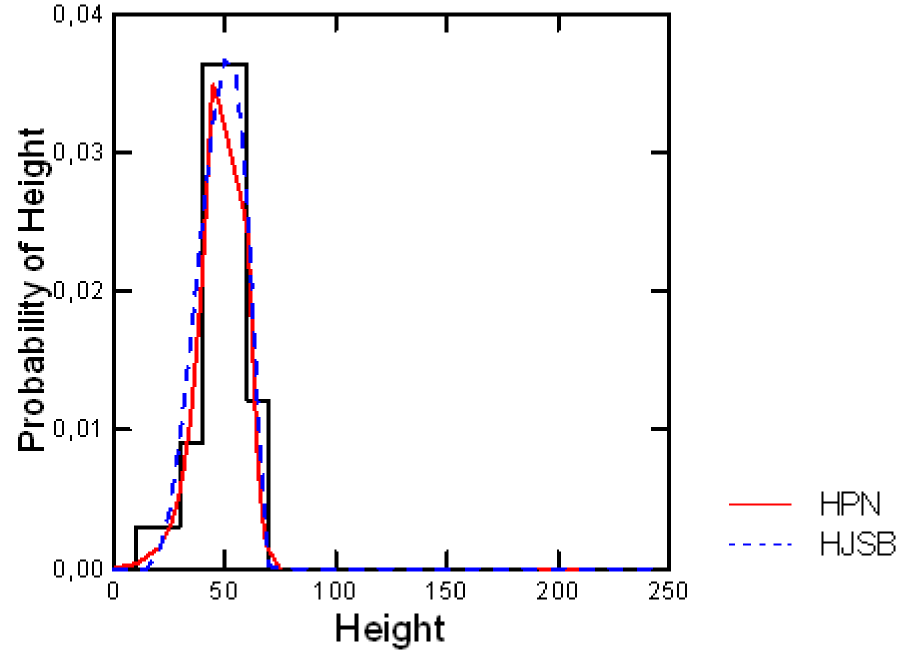


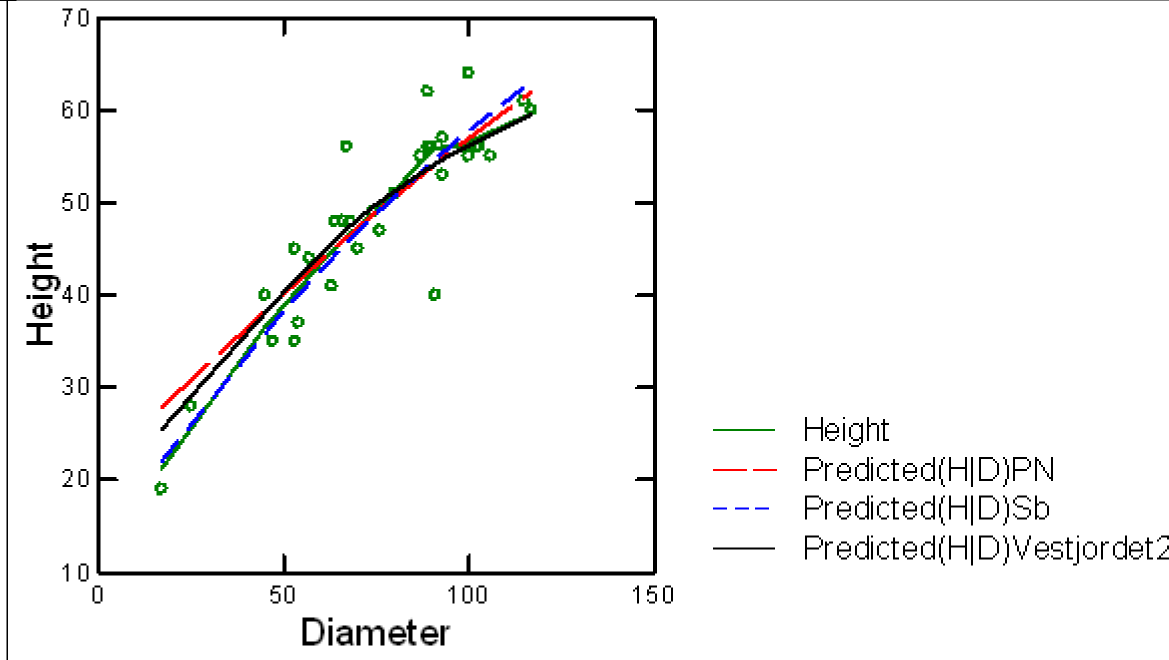




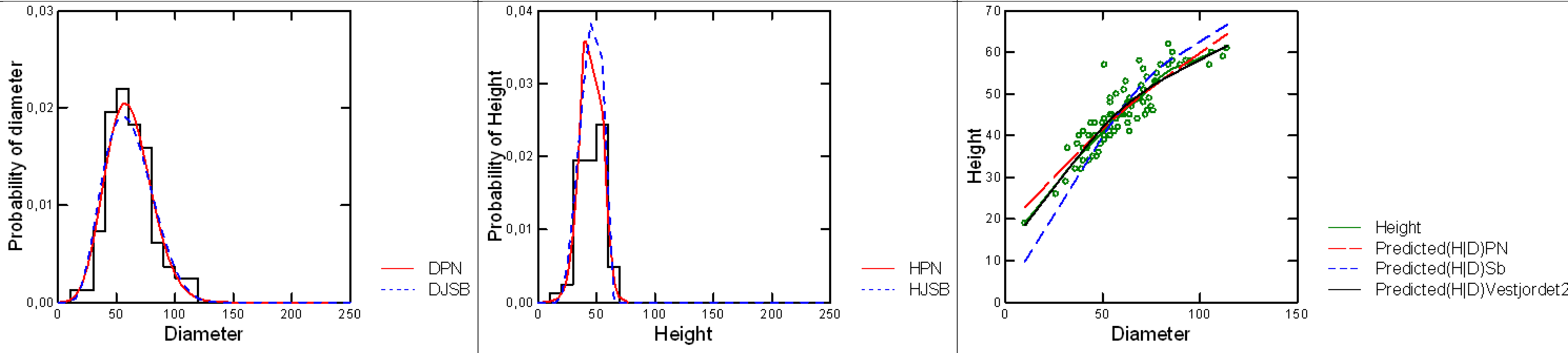
Results (↓) for STAND = 134



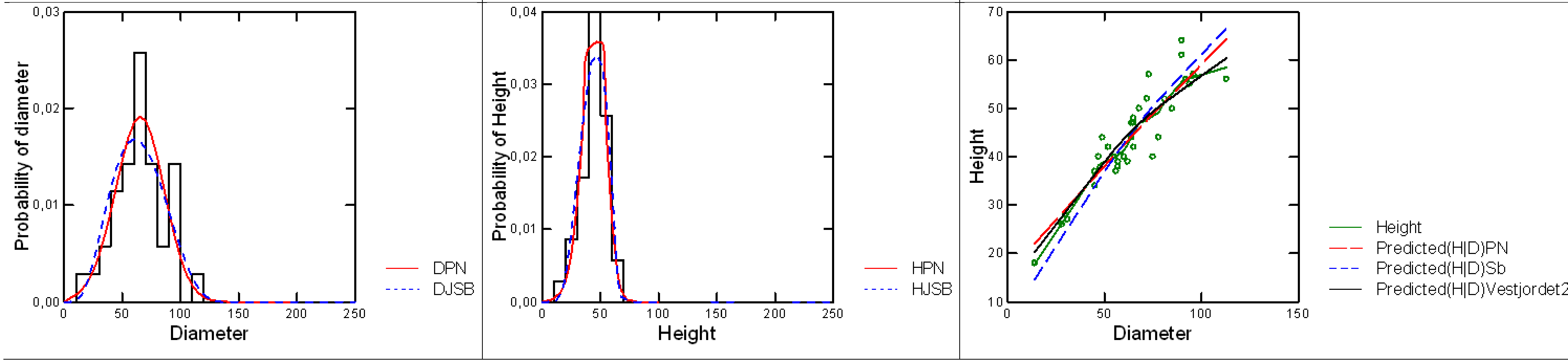




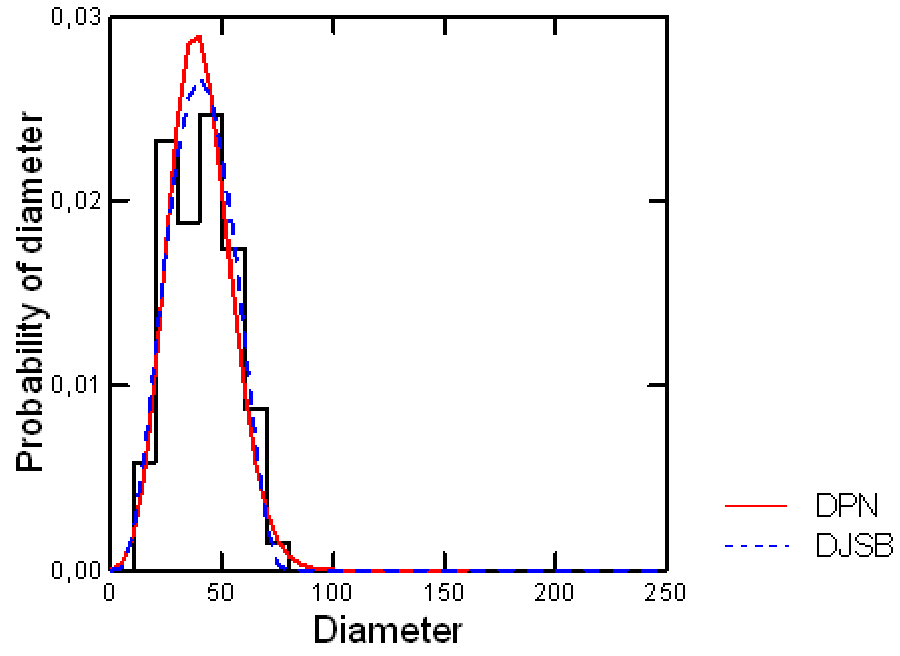
Results (↓) for STAND = 135

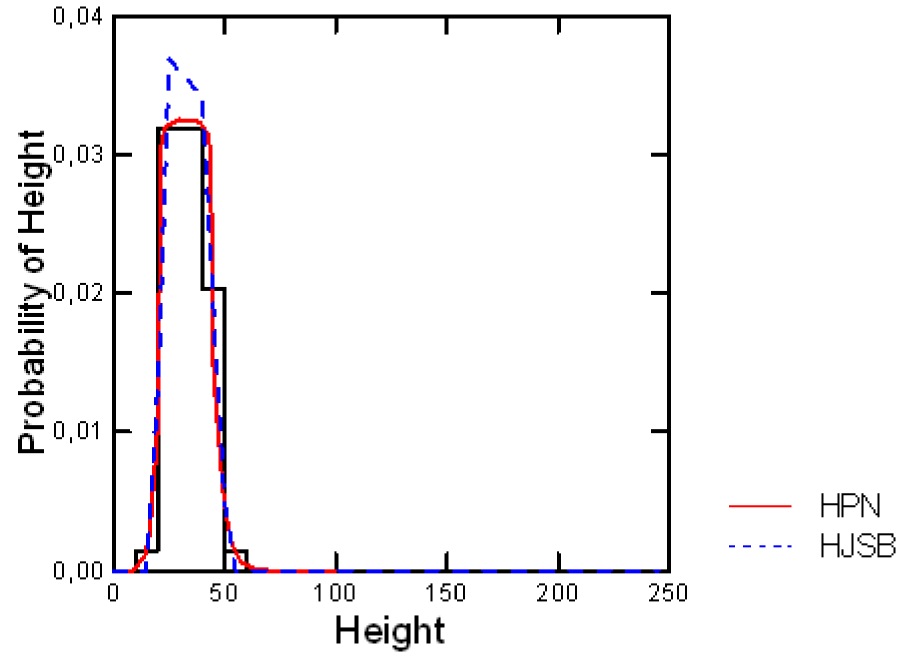


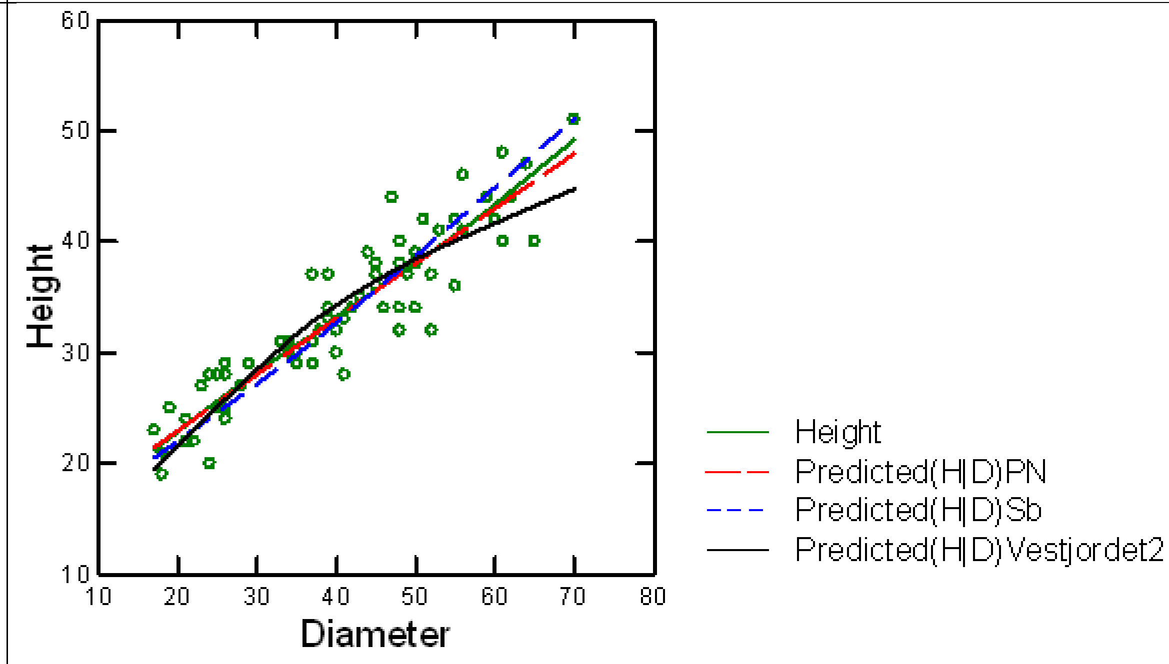
Results (↓) for STAND = 136



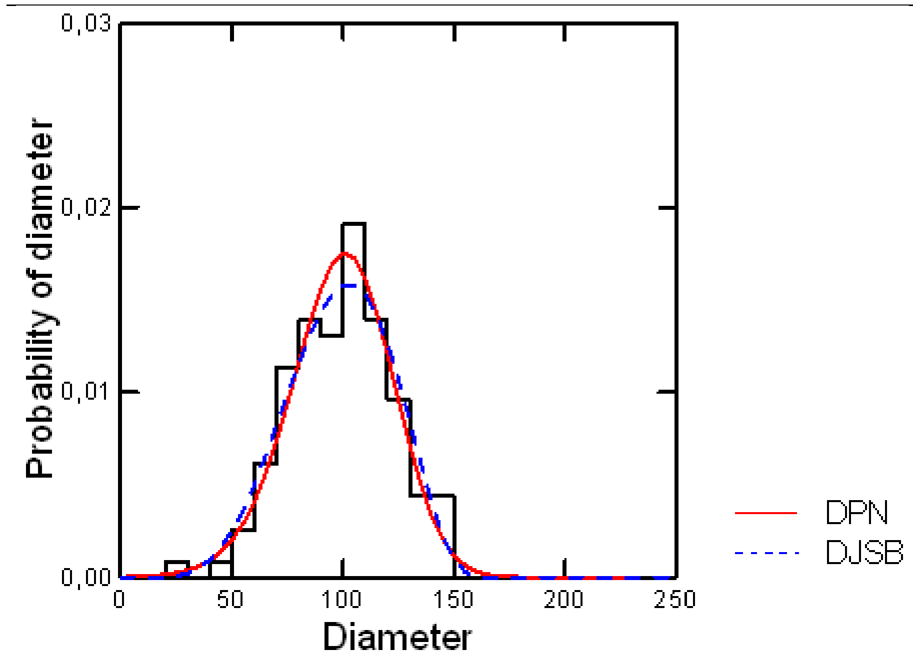
Results (↓) for STAND = 137

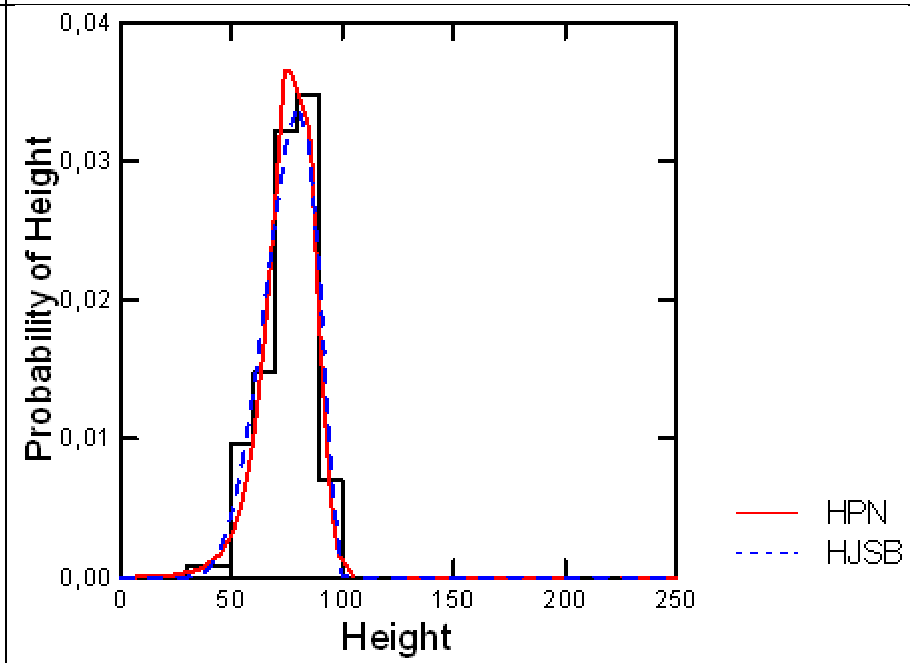


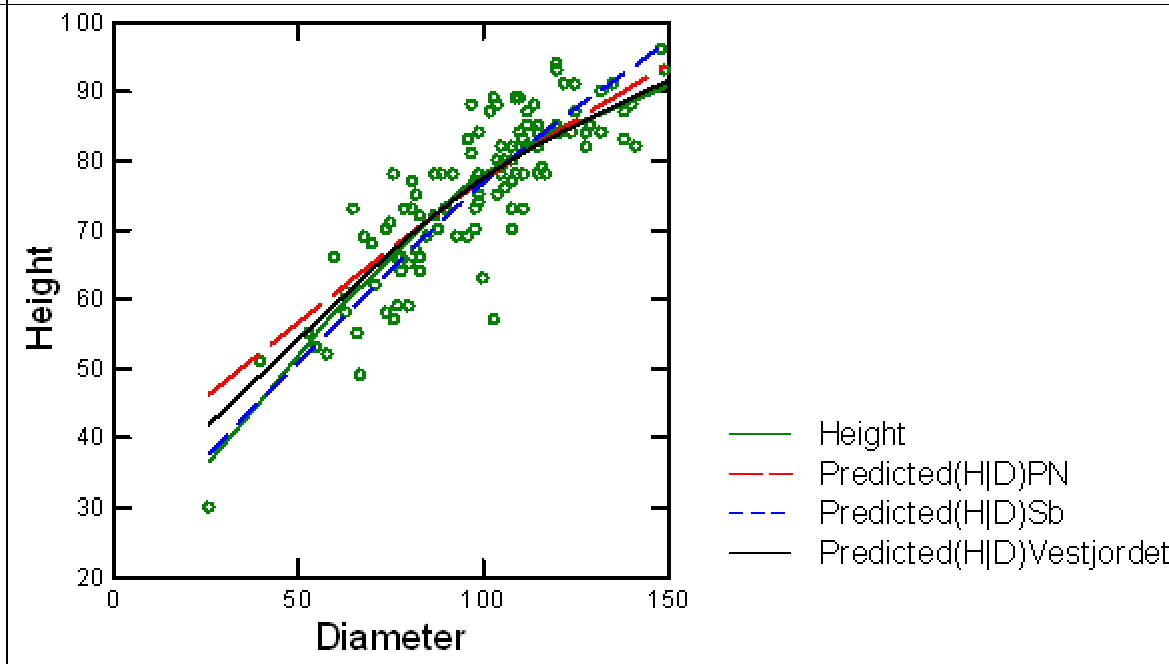




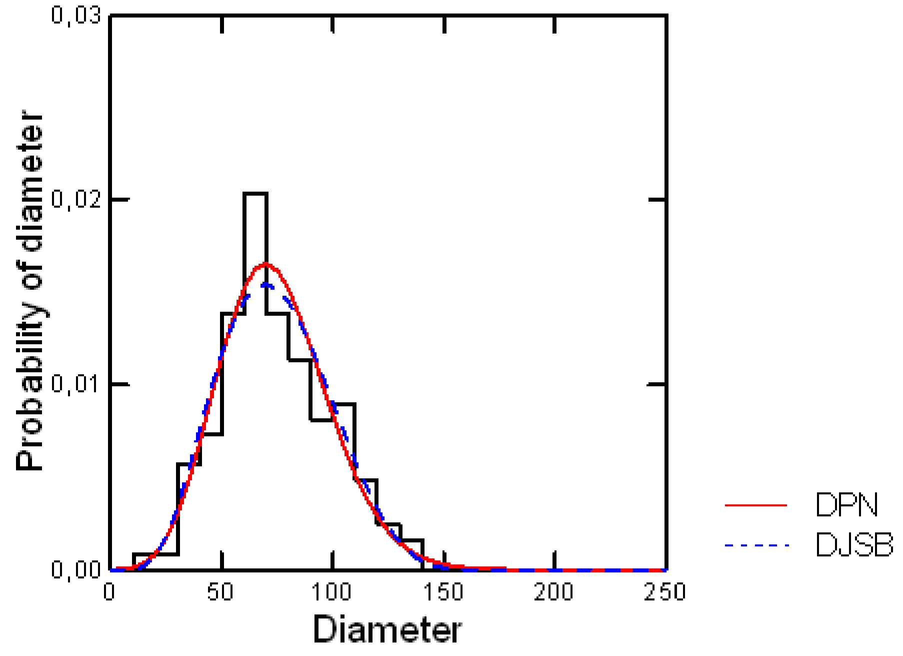
Results (↓) for STAND = 138

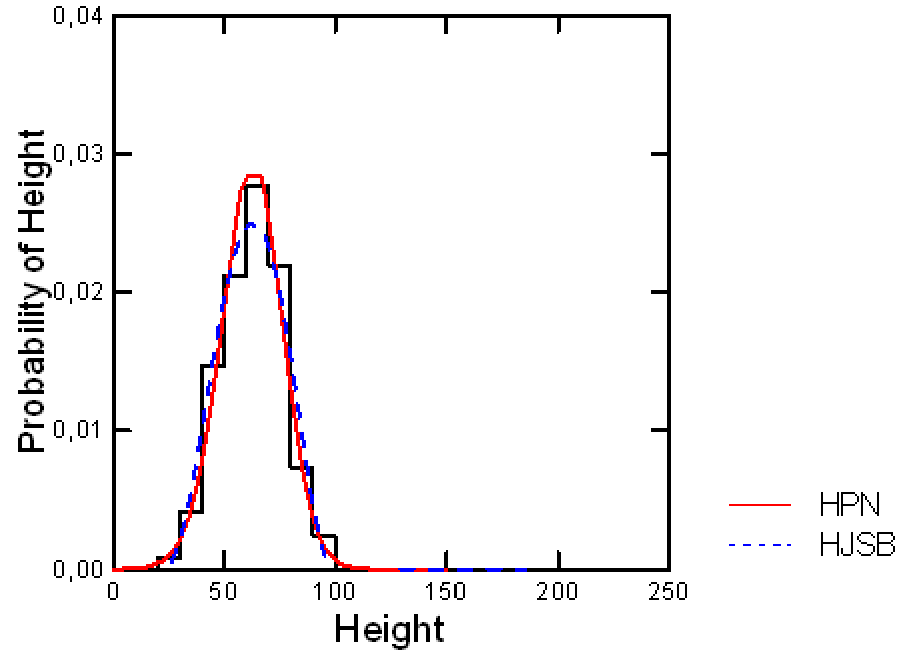


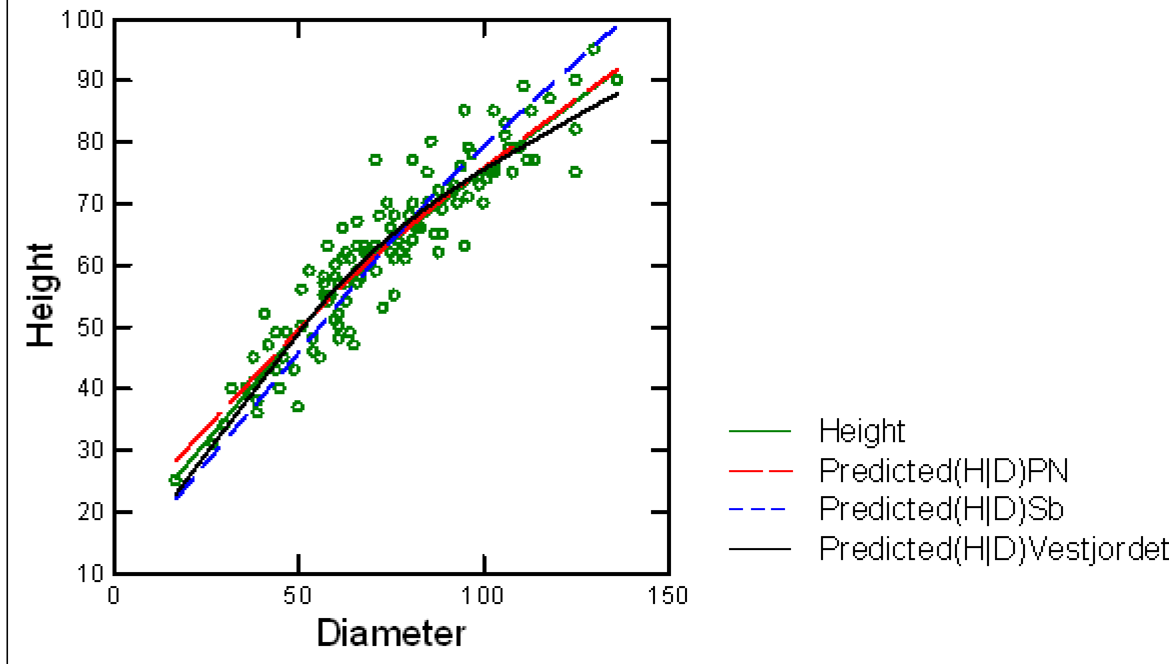




Results (↓) for STAND = 139







Stand characteristics.

STAND	No.Obs.	Dmin	Hmin	Dmax	Hmax	Dg (mm)	Hl (dm)	Dskew	Hskew	Dkurtosis	Hkurtosis
1	48	31	44	160	110	108.35	83.48	-0.16	-0.12	0.05	1.70
2	132	18	30	182	118	94.49	90.97	0.41	-0.67	-0.40	0.62
3	123	18	19	148	87	77.12	63.26	0.29	-0.26	0.30	-0.03
4	217	17	23	152	92	73.75	67.58	0.57	-0.13	0.51	-0.18
5	128	21	27	106	84	57.55	60.35	0.42	0.02	0.37	0.45
6	90	51	51	193	113	105.57	84.94	0.59	0.10	0.05	-0.21
7	101	36	40	149	119	85.08	79.68	0.57	0.21	0.13	0.96
8	80	27	37	164	98	87.78	78.33	0.14	-0.51	-0.14	-0.21
9	87	14	20	106	69	56.23	50.34	0.33	-0.20	0.07	-0.05
10	58	18	21	106	76	61.76	51.66	0.03	0.06	-0.57	0.12
11	50	10	17	96	61	58.47	48.89	-0.03	-0.22	-0.22	-0.02
12	84	4	15	101	73	54.82	50.84	0.24	0.23	-0.35	-0.63
13	93	17	20	112	60	53.45	47.67	0.39	-0.20	0.23	-0.78
14	51	6	15	93	54	46.62	41.96	0.51	0.15	-0.05	-0.56
15	100	88	71	180	128	128.69	108.55	0.21	-0.68	-0.17	1.20
16	189	55	54	153	131	101.71	104.40	0.15	-0.87	-0.41	1.22
17	50	107	85	212	124	153.82	109.41	-0.09	-0.49	0.38	-0.27
18	112	59	57	161	124	120.94	108.44	-0.40	-1.35	-0.31	2.87
19	136	54	60	159	122	109.08	102.96	0.07	-0.88	-0.48	0.69
20	208	40	64	148	118	96.07	101.47	0.18	-0.78	-0.44	0.99
21	63	86	84	208	126	146.55	110.91	0.05	-0.43	-0.10	-0.30
22	106	26	30	116	86	63.81	62.12	0.38	0.12	-0.05	-0.18
23	116	24	29	113	82	72.78	64.77	0.01	-0.72	0.01	0.89
24	116	23	23	102	69	53.79	53.33	0.69	0.17	-0.03	-0.82
25	86	16	22	117	76	63.13	53.76	0.27	-0.24	0.03	0.31
26	89	34	35	115	78	75.34	62.55	-0.05	-0.58	-0.70	0.12
27	74	24	26	113	75	63.62	57.60	0.46	-0.09	0.04	-0.48
28	74	26	27	154	100	76.84	77.84	0.46	-0.30	0.25	0.01
29	95	12	19	145	98	83.29	78.45	0.02	-0.90	-0.53	0.67
30	92	39	49	152	105	89.52	81.82	0.22	-0.25	-0.05	0.16
31	228	19	26	150	75	74.04	58.69	0.43	-0.20	-0.16	-0.09
32	124	19	23	131	71	79.79	56.73	0.02	-0.53	-0.15	0.39
33	79	55	43	169	90	113.56	65.42	0.26	0.05	-0.31	-0.08
34	198	10	14	156	78	75.19	60.03	0.21	-1.03	0.80	2.90
35	122	36	32	152	81	89.62	61.15	-0.01	-0.66	0.00	0.75
36	107	16	23	102	66	61.87	51.29	0.24	-0.13	-0.40	-0.58
37	106	13	26	130	79	89.01	64.76	-0.60	-0.72	-0.21	0.16
38	107	10	20	135	86	78.01	61.52	-0.03	-0.40	-0.77	0.12
39	110	12	22	156	72	81.37	58.02	0.24	-0.82	1.15	1.76
40	108	7	17	133	71	62.16	56.13	0.55	-0.19	-0.39	-0.54
41	108	52	36	136	75	85.19	58.14	0.21	-0.04	0.21	0.49
42	107	24	21	121	69	67.59	52.64	0.25	0.00	-0.49	-0.55
43	85	25	32	159	86	109.78	73.28	-0.75	-1.08	0.93	2.90
44	119	19	27	156	94	93.65	75.08	0.05	-0.50	-0.56	0.01

45	118	21	30	119	74	76.90	59.11	-0.03	-0.13	-0.40	-0.53
46	196	17	22	143	93	77.40	70.50	0.15	-0.55	-0.59	-0.11
47	76	41	36	122	71	85.55	55.81	-0.15	0.00	-0.18	-0.47
48	152	17	23	128	79	76.61	63.07	-0.03	-0.26	-0.47	-0.36
49	96	35	31	134	71	83.84	58.99	0.14	-0.46	0.20	0.27
50	176	19	30	130	104	81.09	82.43	-0.29	-0.75	-0.22	0.09
51	116	45	47	190	120	107.79	99.16	0.34	-0.46	-0.47	-0.61
52	56	64	63	209	122	127.46	95.49	0.31	0.13	0.66	0.22
53	111	64	83	168	131	119.47	112.48	0.16	0.22	-0.64	-0.26
54	154	32	40	188	150	107.50	118.61	0.11	-0.98	-0.34	1.27
55	113	61	68	213	158	131.87	132.57	0.48	-0.58	-0.36	0.73
56	77	58	63	224	148	145.59	126.40	-0.10	-1.03	-0.73	0.85
57	96	47	54	212	152	138.17	125.76	-0.24	-1.20	-0.42	3.31
58	160	28	36	177	124	96.69	100.68	0.17	-1.13	-0.15	1.10
59	162	33	51	158	121	96.12	99.15	0.45	-0.51	-0.37	0.39
60	137	22	28	198	151	110.01	118.60	0.20	-0.81	-0.70	0.02
61	158	16	30	185	140	101.31	107.68	-0.06	-1.03	-0.47	0.41
62	75	32	30	150	100	93.93	74.35	0.24	-0.50	-0.35	1.08
63	78	18	22	126	86	83.48	67.41	-0.37	-1.01	0.38	1.28
64	80	48	45	133	85	90.60	67.50	0.18	-0.06	-0.42	-0.47
65	79	12	19	129	86	78.11	71.42	-0.14	-1.16	0.13	2.14
66	137	41	36	159	118	98.54	87.23	0.24	-0.63	-0.60	0.73
67	114	33	42	182	115	115.68	93.90	-0.14	-0.85	-0.32	1.19
68	98	37	37	211	121	115.05	96.10	0.20	-0.71	0.27	0.94
69	163	17	25	179	121	90.76	91.58	0.33	-0.80	0.09	0.49
70	189	12	21	173	116	81.57	90.22	0.41	-0.27	-0.24	-0.65
71	120	26	35	181	125	98.99	94.84	0.08	-0.59	0.21	0.14
72	63	10	19	175	124	97.57	95.22	0.21	-0.58	0.18	0.21
73	105	13	24	157	107	83.26	82.03	0.30	-0.57	0.29	0.72
74	131	30	37	155	126	89.91	95.43	0.21	-0.63	-0.38	0.43
75	88	37	40	163	130	106.38	99.47	-0.03	-0.48	-0.99	-0.03
76	94	29	34	140	101	86.43	79.94	0.03	-0.96	-0.17	0.91
77	51	25	26	122	82	83.10	66.44	-0.32	-0.82	-0.04	0.34
78	254	16	22	147	87	70.98	66.03	0.09	-0.78	0.50	0.75
79	175	32	33	155	92	84.97	73.10	0.26	-0.48	0.59	0.56
80	129	45	47	167	101	99.50	78.19	0.15	-0.18	0.17	-0.02
81	223	20	28	154	105	80.48	74.11	0.33	-0.51	0.15	0.43
82	161	7	16	156	98	83.64	70.80	0.43	-0.32	0.59	0.52
83	120	7	17	162	98	96.34	73.07	-0.36	-1.06	0.77	2.36
84	235	19	26	124	88	69.80	69.37	0.37	-0.40	-0.29	0.30
85	169	27	37	140	93	84.06	74.94	-0.16	-0.73	-0.06	0.77
86	126	20	21	146	92	88.58	73.25	-0.45	-1.00	0.25	1.53
87	170	16	19	184	110	78.23	76.62	0.85	-0.43	1.12	0.15
88	180	20	27	140	96	85.61	73.15	-0.32	-0.84	-0.19	0.95
89	84	54	42	147	96	97.85	74.23	0.06	-0.20	-0.42	0.64
90	138	27	35	165	117	91.04	81.09	0.25	-0.47	-0.16	0.38

91	180	5	14	172	117	87.95	87.68	0.32	-0.38	-0.67	-0.02
92	148	8	19	182	113	94.98	89.93	0.26	-0.63	-0.45	-0.08
93	341	11	18	183	123	88.44	96.31	0.20	-0.82	-0.34	0.13
94	261	7	16	163	112	86.43	84.40	0.24	-0.38	-0.73	-0.43
95	527	12	19	143	101	67.93	67.01	0.61	0.09	0.09	0.01
96	158	19	22	178	139	90.81	88.87	0.20	-0.25	-0.53	-0.10
97	170	15	18	164	105	84.39	76.80	0.23	-0.73	0.02	1.06
98	77	16	22	206	117	118.47	93.97	-0.18	-0.56	-0.18	0.81
99	122	32	39	176	108	108.85	86.42	-0.15	-0.71	-0.79	0.47
100	126	12	20	146	109	73.54	74.28	0.37	-0.05	-0.48	-1.07
101	85	16	21	171	116	91.40	85.26	0.39	-0.31	-0.41	-0.56
102	79	23	28	137	81	87.21	64.39	0.01	-0.50	-0.24	0.01
103	127	4	12	84	61	49.83	42.45	-0.15	-0.12	-0.34	0.17
104	94	16	21	111	69	66.60	51.50	-0.15	-0.25	-0.29	-0.05
105	114	19	28	113	78	72.06	57.79	-0.17	-0.21	0.34	0.39
106	222	5	16	136	92	70.96	66.90	0.30	-0.36	-0.18	0.23
107	87	52	52	169	95	109.00	78.93	0.33	-0.21	0.75	-0.08
108	112	11	17	117	69	55.33	46.20	0.29	0.16	0.50	-0.01
109	149	8	16	110	84	65.39	59.82	-0.06	-0.39	-0.26	0.25
110	135	3	13	104	76	52.92	52.96	0.29	0.18	-0.64	-0.76
111	77	23	24	95	71	62.23	51.15	0.06	-0.08	-0.60	-0.13
112	51	33	32	121	70	70.47	54.16	0.41	0.35	-0.31	-0.72
113	162	13	19	124	70	60.81	53.02	0.37	-0.30	-0.41	-0.67
114	106	10	17	110	70	64.46	50.68	0.01	0.01	-0.68	-0.49
115	193	5	15	127	80	60.29	56.12	0.15	-0.49	-0.29	-0.10
116	78	28	30	150	86	91.13	66.48	-0.15	-0.41	-0.03	-0.16
117	143	9	17	133	88	78.94	63.04	-0.01	-0.12	-0.57	-0.19
118	102	13	18	108	63	63.48	49.59	0.21	-0.17	-0.21	-0.27
119	50	28	26	92	61	63.90	47.32	0.01	-0.01	-0.48	0.41
120	60	25	27	122	80	76.72	60.58	0.04	0.00	-0.66	-0.64
121	174	8	17	114	86	58.23	59.21	0.30	-0.09	-0.35	-0.29
122	121	15	23	140	89	80.54	71.27	-0.18	-0.84	-0.01	0.53
123	143	14	20	143	78	63.34	51.30	0.77	0.26	1.73	-0.14
124	53	40	31	179	77	89.10	58.42	0.91	0.10	2.35	-0.63
125	76	22	26	143	83	84.46	64.35	-0.22	-1.11	-0.23	1.26
126	53	39	37	126	73	84.20	59.87	-0.04	-0.14	-0.39	-0.59
127	142	8	17	133	75	62.22	58.14	0.24	-0.12	-0.37	-0.88
128	162	15	22	159	110	79.60	82.22	0.31	-0.26	-0.31	0.01
129	66	28	30	141	102	97.49	80.01	-0.23	-0.47	-0.55	1.55
130	121	14	21	141	94	79.57	70.98	-0.06	-0.51	-0.59	-0.08
131	90	24	27	148	88	78.63	62.75	0.44	0.05	-0.04	-0.60
132	157	6	15	155	100	80.47	76.48	0.10	-0.65	-0.51	0.09
133	94	24	25	133	91	83.79	68.29	0.01	-0.38	-0.66	0.13
134	33	17	19	117	64	79.85	53.12	-0.44	-0.91	-0.23	0.84
135	82	10	19	114	62	63.70	50.09	0.50	-0.19	0.51	-0.10
136	35	14	18	113	64	67.76	49.15	-0.12	-0.33	0.20	0.17

137	69	17	19	70	51	42.45	37.38	0.07	0.17	-0.93	-0.57
138	115	26	30	149	96	101.27	79.59	-0.26	-0.90	0.08	1.28
139	123	17	25	136	95	77.89	70.02	0.30	-0.12	-0.35	-0.25

SAS programs: Two-dimensional Kolmogorov-Smirnov in 4 versions

$P\left(D \le d \bigcap H \le h\right)$

```
            /* 2014 juli Beregning Bivariat KolmogorovSmirnov
                                                H< og d<
             FIL=PN_JSB_KS-2D          */
options ps=63;
data hoydiam.dhKolmogorovSmirnov2D;
merge hoydiam.dhhorisontal hoydiam.dhbox1 hoydiam.dhparjsbf; by flate;

array dd{550} 3 dd1-dd550;              /*  enkelt-dataverdier */
array hh{550} 3 hh1-hh550;              /*  enkelt-dataverdier */
array ant{550,550} _temporary_;
                              /* beregn for en flate */
drop dd1-dd550 hh1-hh550 k l I ld lh distpn distjsb diff;
                                                          /* if flate=1 then DO */
DO k=1 TO n;
DO l=1 TO n;
        ant{k,l}=0; distjsb=0; distpn=0;
        DO i=1 TO n;
        IF ((dd{i}<=dd{k}) AND (hh{i}<=hh{l})) then ant{k,l}=ant{k,l}+1;
        End;
ant{k,l}= ant{k,l}/n;
                                  /* Beregne Kolomgorov-Smirov for SJB */
ld=(log((dd{k}-dtau)/(dteta-dd{k}))-dmy)/dbeta;
lh=(log((hh{l}-htau)/(hteta-hh{l}))-hmy)/hbeta;

diff= ABS(PROBBNRM(ld, lh, rojsb)- ant{k,l});
IF (distjsb < diff) then distjsb = diff;
                                  /* Beregne Kolomgorov-Smirov for PN */
ld=(dd{k}**dlanda-(1+dlanda*dbmy))/(dlanda*dbsigma);
lh=(hh{l}**hlanda-(1+hlanda*hbmy))/(hlanda*hbsigma);
KK=1;
        if (dlanda>0) AND (hlanda>0) then
         KK=PROBBNRM((1+dlanda*dbmy)/(dlanda*dbsigma),
                (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else

        if (dlanda<0) AND (hlanda<0) then
         KK=PROBBNRM(-(1+dlanda*dbmy)/(dlanda*dbsigma),
                - (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else
        if (dlanda<0) AND (hlanda>0) then
         KK=PROBBNRM(-(1+dlanda*dbmy)/(dlanda*dbsigma),
                (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else
        if (dlanda>0) AND (hlanda<0) then
```

```
        KK=PROBBNRM((1+dlanda*dbmy)/(dlanda*dbsigma),
                - (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);

diff= ABS(PROBBNRM(ld, lh, ropn)/KK- ant{k,l});
IF (distpn < diff) then distpn = diff;
End; End;
KSjsb= distjsb ;
KSpn= distpn ;
Put 'flate= ' flate ksjsb kspn;
/*End*/
Run;
```

$P\left(D \le d \bigcap H \ge h\right)$

```
            /* 2014 juli Beregning Bivariat KolmogorovSmirnov
                        Definert som øvre venstre kvadrant
                         h> og d<
             FIL=PN_JSB_KS-2D-II           */
options ps=63;
data hoydiam.dhKolmogorovSmirnov2D_II;
merge hoydiam.dhhorisontal hoydiam.dhbox1 hoydiam.dhparjsbf; by flate;

array dd{550} 3 dd1-dd550;              /*  enkelt-dataverdier */
array hh{550} 3 hh1-hh550;              /*  enkelt-dataverdier */
array ant{550,550} _temporary_;
                                 /* beregn for en flate */
drop dd1-dd550 hh1-hh550 k l I ld lh distpn distjsb diff;
                                                                  /* if flate=1 then DO */
DO k=1 TO n;
DO l=1 TO n;
        ant{k,l}=0; distjsb=0; distpn=0;
        DO i=1 TO n;
        IF ((dd{i}<=dd{k}) AND (hh{i}>=hh{l})) then ant{k,l}=ant{k,l}+1;
        End;
ant{k,l}= ant{k,l}/n;
                                        /* Beregne Kolomgorov-Smirov for SJB */
ld=(log((dd{k}-dtau)/(dteta-dd{k}))-dmy)/dbeta;
lh=(log((hh{l}-htau)/(hteta-hh{l}))-hmy)/hbeta;

diff= ABS((PROBBNRM(ld,1000,rojsb)-PROBBNRM(ld, lh, rojsb))- ant{k,l});
IF (distjsb < diff) then distjsb = diff;
                                        /* Beregne Kolomgorov-Smirov for PN */
ld=(dd{k}**dlanda-(1+dlanda*dbmy))/(dlanda*dbsigma);
lh=(hh{l}**hlanda-(1+hlanda*hbmy))/(hlanda*hbsigma);
KK=1;
        if (dlanda>0) AND (hlanda>0) then
         KK=PROBBNRM((1+dlanda*dbmy)/(dlanda*dbsigma),
                (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else

        if (dlanda<0) AND (hlanda<0) then
         KK=PROBBNRM(-(1+dlanda*dbmy)/(dlanda*dbsigma),
                - (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
```

```
else
            if (dlanda<0) AND (hlanda>0) then
             KK=PROBBNRM(-(1+dlanda*dbmy)/(dlanda*dbsigma),
                      (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else
            if (dlanda>0) AND (hlanda<0) then
             KK=PROBBNRM((1+dlanda*dbmy)/(dlanda*dbsigma),
                      - (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);

diff= ABS(PROBBNRM(ld, 1000, ropn)/KK -PROBBNRM(ld, lh, ropn)/KK- ant{k,l});
IF (distpn < diff) then distpn = diff;
End; End;
KSjsb= distjsb ;
KSpn= distpn ;
Put 'flate= ' flate ksjsb kspn;
/*End*/
Run;
```

$P\left(D \geq d \cap H \leq h\right)$

```
                /* 2014 juli Beregning Bivariat KolmogorovSmirnov
                            Definert som nedre høyre kvadrant
                             H< og d>
                  FIL=PN_JSB_KS-2D-III           */
options ps=63;
data hoydiam.dhKolmogorovSmirnov2D_III;
merge hoydiam.dhhorisontal hoydiam.dhbox1 hoydiam.dhparjsbf; by flate;

array dd{550} 3 dd1-dd550;                /*  enkelt-dataverdier */
array hh{550} 3 hh1-hh550;                /*  enkelt-dataverdier */
array ant{550,550} _temporary_;
                                      /* beregn for en flate */
drop dd1-dd550 hh1-hh550 k l I ld lh distpn distjsb diff;
                                                                        /* if flate=1 then DO */
DO k=1 TO n;
DO l=1 TO n;
            ant{k,l}=0; distjsb=0; distpn=0;
            DO i=1 TO n;
            IF ((dd{i}>=dd{k}) AND (hh{i}<=hh{l})) then ant{k,l}=ant{k,l}+1;
            End;
ant{k,l}= ant{k,l}/n;
                                              /* Beregne Kolomgorov-Smirov for SJB */
ld=(log((dd{k}-dtau)/(dteta-dd{k}))-dmy)/dbeta;
lh=(log((hh{l}-htau)/(hteta-hh{l}))-hmy)/hbeta;

diff= ABS((PROBBNRM(1000,lh,rojsb)-PROBBNRM(ld, lh, rojsb))- ant{k,l});
IF (distjsb < diff) then distjsb = diff;
                                              /* Beregne Kolomgorov-Smirov for PN */
ld=(dd{k}**dlanda-(1+dlanda*dbmy))/(dlanda*dbsigma);
lh=(hh{l}**hlanda-(1+hlanda*hbmy))/(hlanda*hbsigma);
KK=1;
            if (dlanda>0) AND (hlanda>0) then
             KK=PROBBNRM((1+dlanda*dbmy)/(dlanda*dbsigma),
```

```
                    (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else

           if (dlanda<0) AND (hlanda<0) then
            KK=PROBBNRM(-(1+dlanda*dbmy)/(dlanda*dbsigma),
                    - (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else
           if (dlanda<0) AND (hlanda>0) then
            KK=PROBBNRM(-(1+dlanda*dbmy)/(dlanda*dbsigma),
                    (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else
           if (dlanda>0) AND (hlanda<0) then
            KK=PROBBNRM((1+dlanda*dbmy)/(dlanda*dbsigma),
                    - (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);

diff= ABS(PROBBNRM(1000,lh, ropn)/KK -PROBBNRM(ld, lh, ropn)/KK- ant{k,l});
IF (distpn < diff) then distpn = diff;
End; End;
KSjsb= distjsb ;
KSpn= distpn ;
Put 'flate= ' flate ksjsb kspn;
/*End*/
Run;
```

$P\left(D \geq d \cap H \geq h\right)$

```
              /* 2014 juli Beregning Bivariat KolmogorovSmirnov
                         Definert som øvre høyre kvadrant
                          h> og d>
               FIL=PN_JSB_KS-2D-IV           */
options ps=63;
data hoydiam.dhKolmogorovSmirnov2D_IV;
merge hoydiam.dhhorisontal hoydiam.dhbox1 hoydiam.dhparjsbf; by flate;

array dd{550} 3 dd1-dd550;                /*  enkelt-dataverdier */
array hh{550} 3 hh1-hh550;                /*  enkelt-dataverdier */
array ant{550,550} _temporary_;
                                  /* beregn for en flate */
drop dd1-dd550 hh1-hh550 k l I ld lh distpn distjsb diff;
                                                               /* if flate=1 then DO */
DO k=1 TO n;
DO l=1 TO n;
           ant{k,l}=0; distjsb=0; distpn=0;
           DO i=1 TO n;
           IF ((dd{i}>=dd{k}) AND (hh{i}>=hh{l})) then ant{k,l}=ant{k,l}+1;
           End;
ant{k,l}= ant{k,l}/n;
                                        /* Beregne Kolomgorov-Smirov for SJB */
ld=(log((dd{k}-dtau)/(dteta-dd{k}))-dmy)/dbeta;
lh=(log((hh{l}-htau)/(hteta-hh{l}))-hmy)/hbeta;

diff= ABS(1-(PROBBNRM(1000,lh,rojsb)+ PROBBNRM(ld, 1000,rojsb)-
                    PROBBNRM(ld, lh, rojsb))- ant{k,l});
```

```
IF (distjsb < diff) then distjsb = diff;
                                        /* Beregne Kolomgorov-Smirov for PN */
ld=(dd{k}**dlanda-(1+dlanda*dbmy))/(dlanda*dbsigma);
lh=(hh{l}**hlanda-(1+hlanda*hbmy))/(hlanda*hbsigma);
KK=1;
          if (dlanda>0) AND (hlanda>0) then
           KK=PROBBNRM((1+dlanda*dbmy)/(dlanda*dbsigma),
                  (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else

          if (dlanda<0) AND (hlanda<0) then
           KK=PROBBNRM(-(1+dlanda*dbmy)/(dlanda*dbsigma),
                  - (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else
          if (dlanda<0) AND (hlanda>0) then
           KK=PROBBNRM(-(1+dlanda*dbmy)/(dlanda*dbsigma),
                  (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);
else
          if (dlanda>0) AND (hlanda<0) then
           KK=PROBBNRM((1+dlanda*dbmy)/(dlanda*dbsigma),
                  - (1+hlanda*hbmy)/(hlanda*hbsigma),ropn);

diff= ABS(1-(PROBBNRM(1000,lh, ropn)/KK+ PROBBNRM(ld,1000, ropn)/KK -PROBBNRM(ld, lh, ropn)/KK)- ant{k,l});
IF (distpn < diff) then distpn = diff;
End; End;
KSjsb= distjsb ;
KSpn= distpn ;
Put 'flate= ' flate ksjsb kspn;
/*End*/
Run;
```