

Seno krasta līniju detektēšana, izmantojot brīvpieejas rīkus

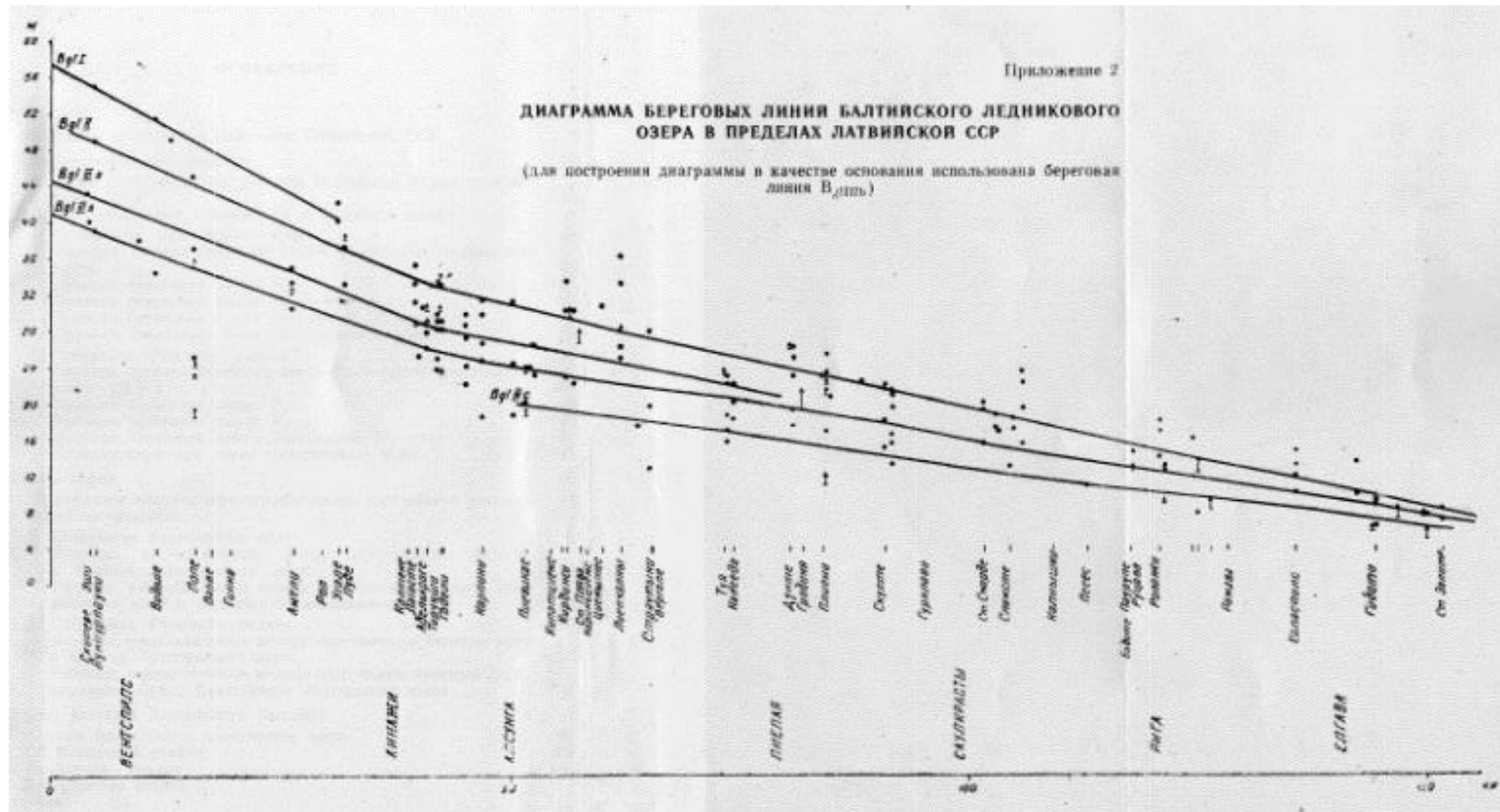
Edijs Breijers

LU Ģeogrāfijas un Zemes zinātņu fakultāte

e-pasts: ebreijers@gmail.com

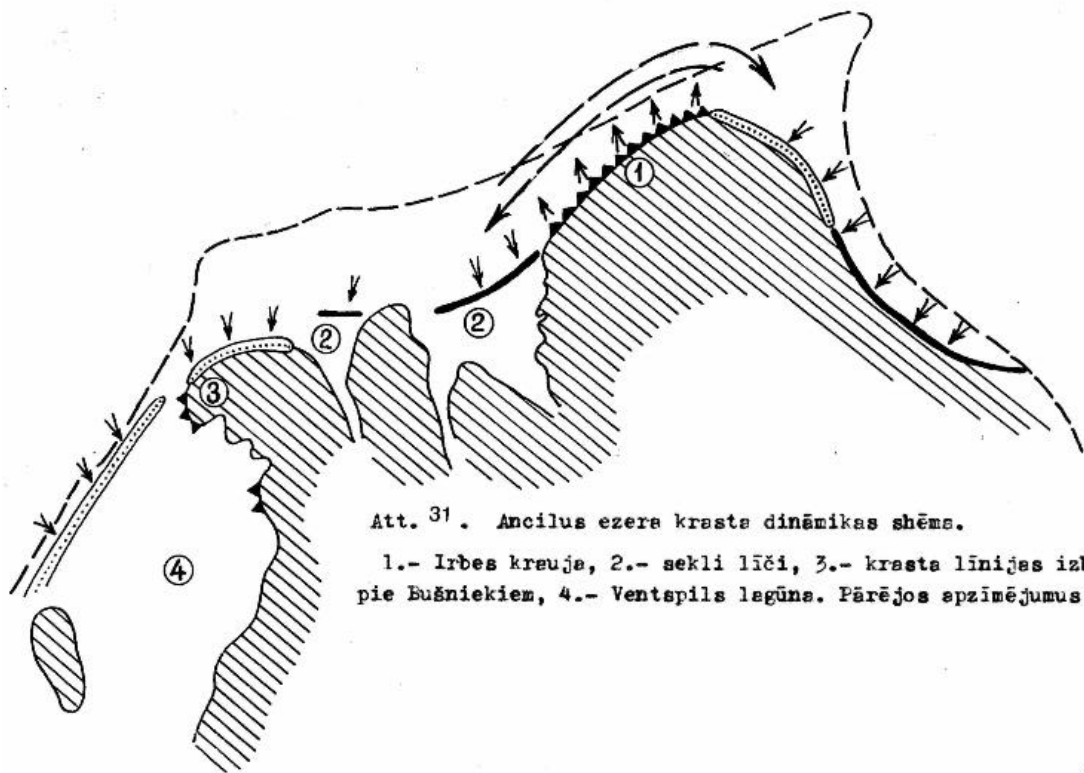
Seno krasta līniju pētījumi Latvijā

Grīnbergs 1957. gadā



Seno krasta līniju pētījumi Latvijā

Veinbergs 1996. gadā



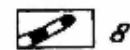
Att. 31. Ancilus ezera krasta dināmas shēma.

1.- Irbes kreuzs, 2.- sekli līči, 3.- krasta līnijas izbīdījums pie Bušniekiem, 4.- Ventpils lagūna. Pārējos apzīmējumus sk. att. 9

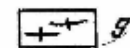


Att. 32. Baltijas jūras piekrastes Litorīnas jūras (fāze Lit_g) krasta zonas dināmas shēma.

1.- Nidas lagūna, 2.- Papes lagūna, 3.- Liepājas lagūna, 4.- Pāvilostas - Jūrkaines abrāzītes krasts, 5.- Ventpils lagūna, 6.- Kurzemes pussēdas ziemeļdaļes akumulatīvais veidojums, 7.- Baltijas jūras austrumdaļes sanesu plūsma, 8.- pāržmeugss, 9.- vaļņveida kāpu ģenerācijas.



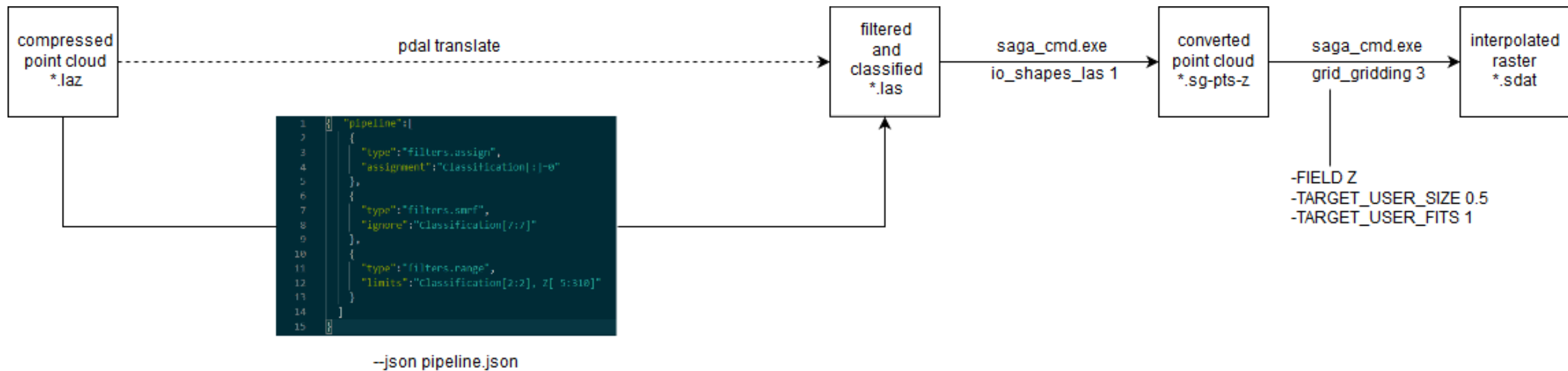
8



9

Seno krasta līniju pētījumu pamats

Digitālais reljefa modelis, kas veidots no LĢIA aerolāzerskenēšanas datiem



```

1 import urllib.request
2 import os
3 import subprocess as sub
4 from tqdm import tqdm
5
6 karsu_lapas = open(r"X:\ceļš\uz\tks92_2k\lapām.txt", "r")
7 path = r"X:\ceļš\uz\LAS"
8 out_path = r"X:\ceļš\uz\DEM"
9
10 class DownloadProgressBar(tqdm):
11     def update_to(self, b=1, bsize=1, tsize=None):
12         if tsize is not None:
13             self.total = tsize
14             self.update(b * bsize - self.n)
15
16 def download_url(url, filename):
17     with DownloadProgressBar(unit='B', unit_scale=True,
18                             miniters=1, desc=url.split('/')[-1]) as t:
19         urllib.request.urlretrieve(url, filename=las_name, reporthook=t.update_to)
20
21 for lapa in karsu_lapas.readlines():
22     url = "http://s3.storage.pub.lvdc.gov.lv/lgia-opendata/las/" + lapa[0:4] + "/" + lapa[0:10] + ".las"
23     las_name = path + "\\" + lapa[0:10] + ".las"
24     download_url(url, las_name)
25
26     newname_pdal = os.path.join(out_path + "\\" + lapa[0:10] + '.las')
27     sauciens_pdal = [r"C:\OSGeo4W\OSGeo4W.bat",
28                     "pdal", "translate", las_name, newname_pdal,
29                     "--json", r"X:\ceļš\uz\pipeline_assign_smr.json"
30                     ]
31
32     print ('Iegūst {0} zemes punktus.'.format(las_name))
33     p = sub.Popen(sauciens_pdal, stdout=sub.PIPE, stderr=sub.PIPE, shell=True)
34     stdout, stderr = p.communicate()
35     if p.returncode != 0:
36         print (stdout)
37         print (stderr)
38
39     newname_sagapts = os.path.join(out_path + "\\" + lapa[0:10] + '.sg-pts-z')
40     sauciens_sagapts = [r"C:\SAGA-GIS\saga_cmd.exe",
41                         "io_shapes_las", "1", "-FILES", newname_pdal, "-POINTS", newname_sagapts, "-i", "1"
42                         ]
43
44

```

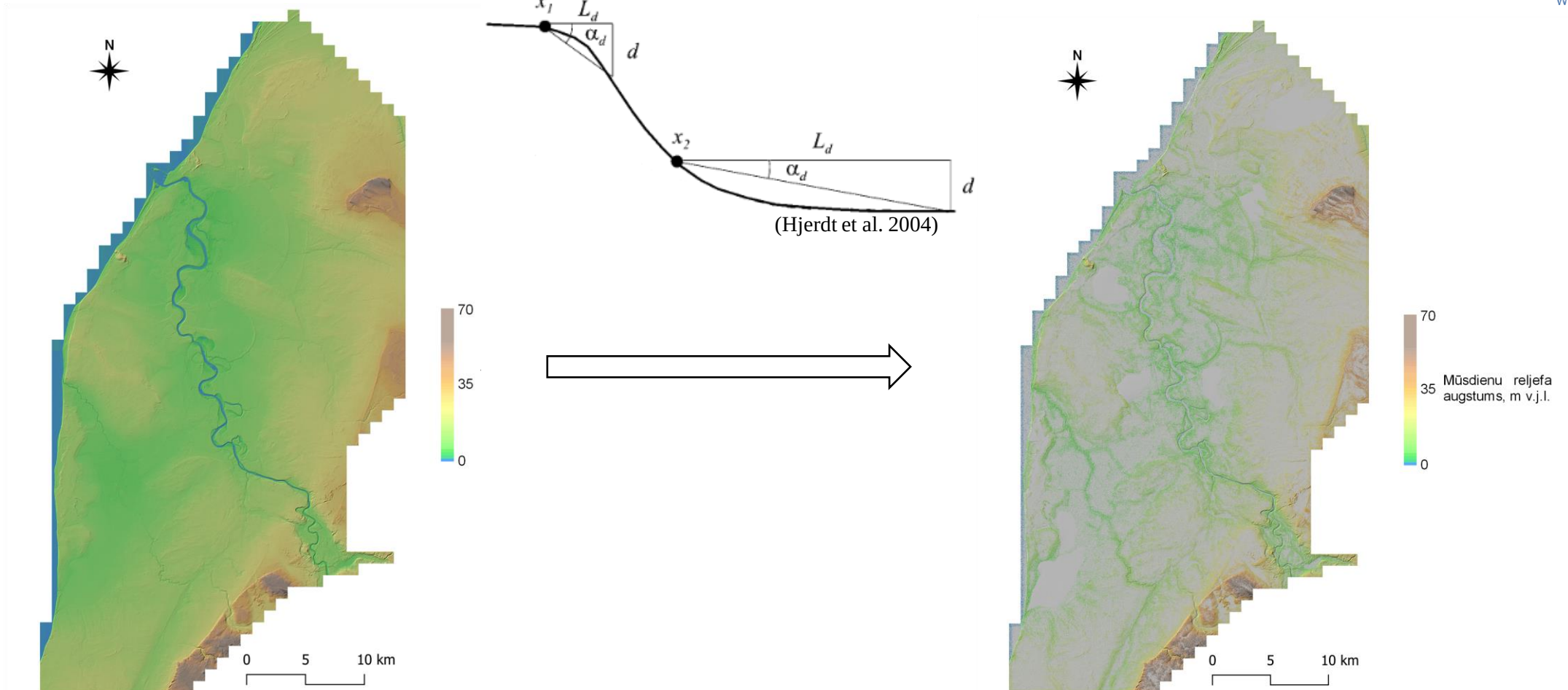
```

45     print ('Konvertē {0} uz {1}'.format(newname_pdal, newname_sagapts))
46     p = sub.Popen(sauciens_sagapts, stdout=sub.PIPE, stderr=sub.PIPE)
47     stdout, stderr = p.communicate()
48     if p.returncode != 0:
49         print (stdout)
50         print (stderr)
51
52     newname_sagagrid = os.path.join(out_path + "\\" + lapa[0:10] + '.sgrd')
53     sauciens_sagagrid = [r"C:\SAGA-GIS\saga_cmd.exe",
54                          "grid_gridding", "3", "-POINTS", newname_sagapts, "-FIELD", "Z", "-TARGET_USER_SIZE",
55                          "5", "-TARGET_USER_FITS", "1", "-TARGET_OUT_GRID", newname_sagagrid
56                          ]
57
58     print ('Interpolē {0} uz {1}'.format(newname_sagapts, newname_sagagrid))
59     p = sub.Popen(sauciens_sagagrid, stdout=sub.PIPE, stderr=sub.PIPE)
60     stdout, stderr = p.communicate()
61     if p.returncode != 0:
62         print (stdout)
63         print (stderr)
64
65     newname_tif = os.path.join(out_path + "\\" + lapa[0:10] + '.tif')
66     newname_sagadat = os.path.join(out_path + "\\" + lapa[0:10] + '.sdat')
67     sauciens_tif = [r"C:\OSGeo4W\OSGeo4W.bat",
68                    "gdal_translate", "-of", "GTiff", newname_sagadat, newname_tif,
69                    ]
70
71     print ('Konvertē {0} uz {1}'.format(newname_sagadat, newname_tif))
72     p = sub.Popen(sauciens_tif, stdout=sub.PIPE, stderr=sub.PIPE, shell=True)
73     stdout, stderr = p.communicate()
74     if p.returncode != 0:
75         print (stdout)
76         print (stderr)
77     try:
78         os.remove(newname_sagapts)
79         os.remove(newname_sagagrid)
80         os.remove(newname_sagadat)
81         os.remove(newname_pdal)
82         os.remove(os.path.join(out_path + "\\" + lapa[0:10] + '.mgrd'))
83         os.remove(os.path.join(out_path + "\\" + lapa[0:10] + '.prj'))
84         os.remove(las_name)
85     except FileNotFoundError:
86         print('Kļūda!')
87     print(lapa, " gatavs")

```

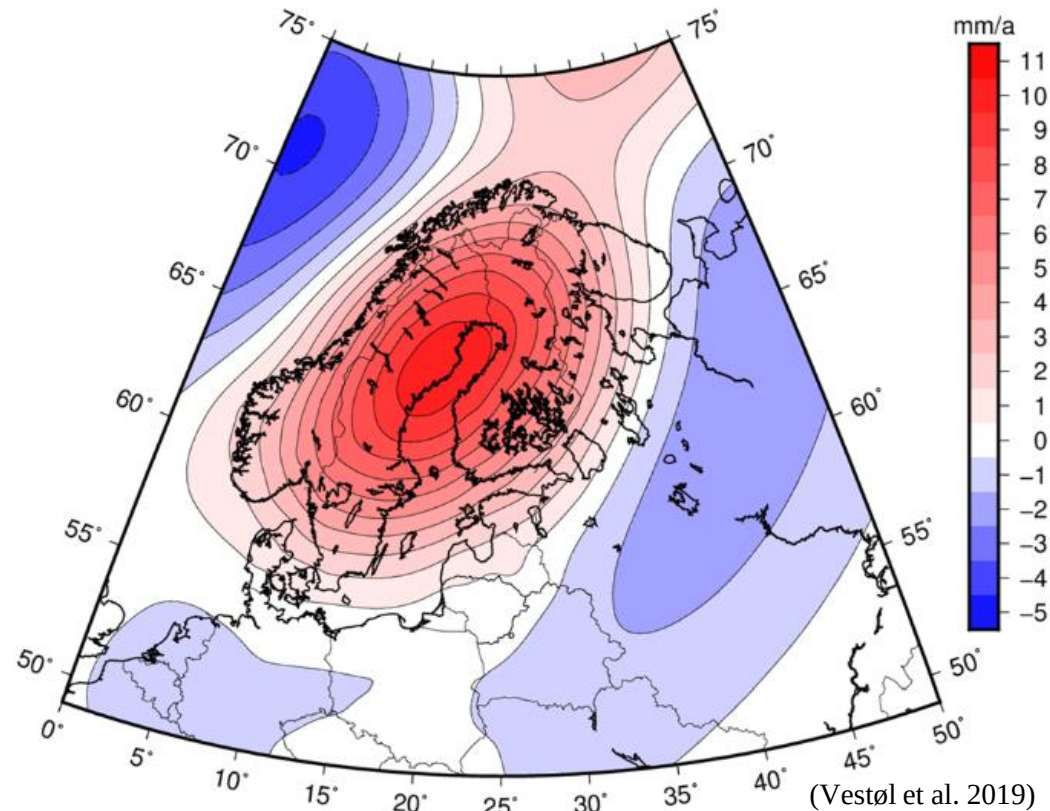
Modelēšanas izejas dati

1. Krasta kāpļu noteikšana – *WhiteboxTools* rīks *DownslopeIndex*



Modelēšanas izejas dati

1. Krasta kāpņu noteikšana – *WhiteboxTools* rīks *DownslopeIndex*
2. Krasta kāpņu pielāgošana glacioizostāzijai – *QGIS/GDAL* rīki



Modelēšanas izejas dati

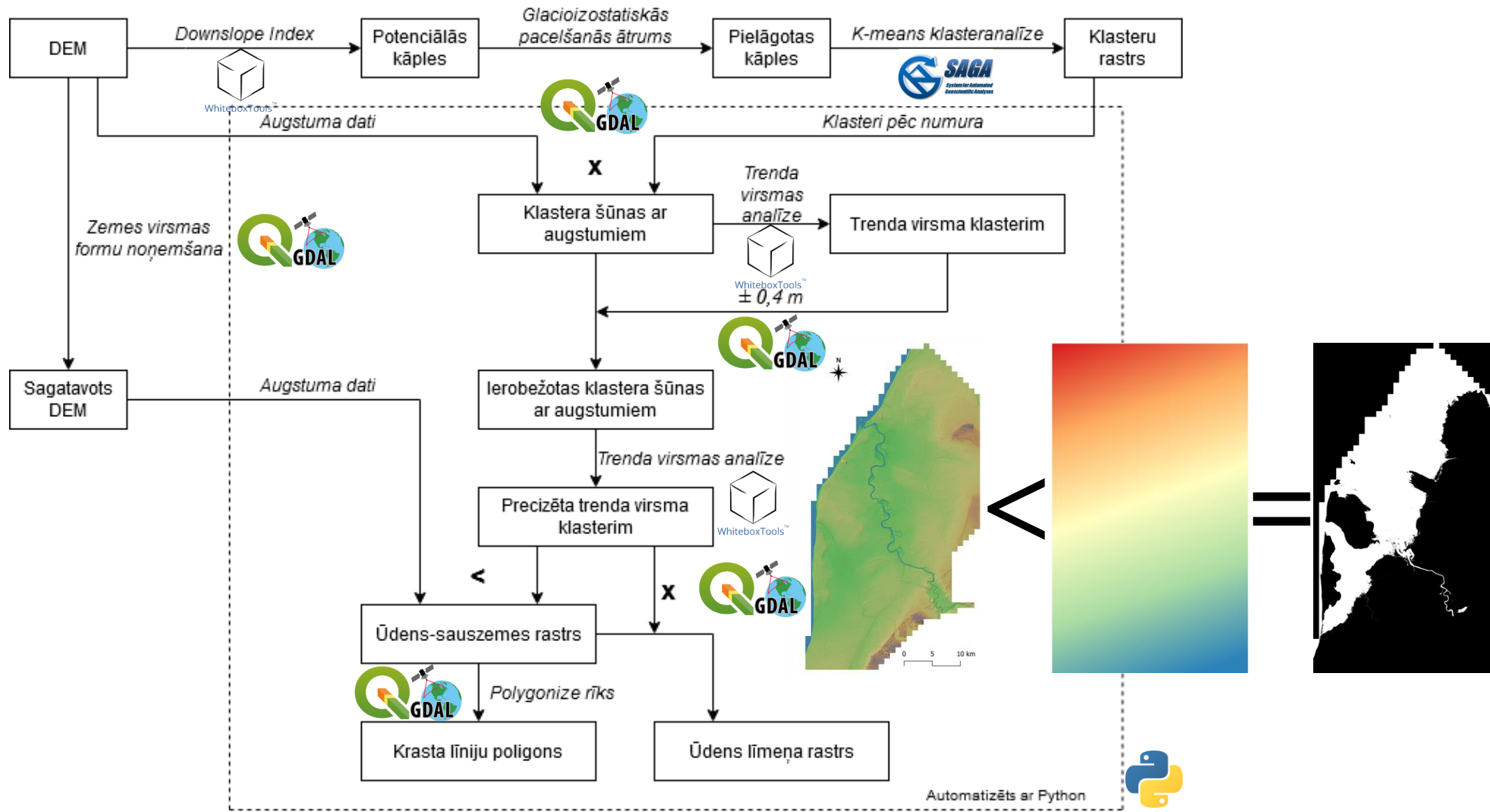
1. Krasta kāpļu noteikšana – *WhiteboxTools* rīks *DownslopeIndex*
2. Krasta kāpļu pielāgošana glacioizostāzijai – *QGIS/GDAL* rīki
3. Pielāgoto krasta kāpļu klasteranalīze – *SAGA GIS* (*K-means clustering for grids*)



WhiteboxTools™

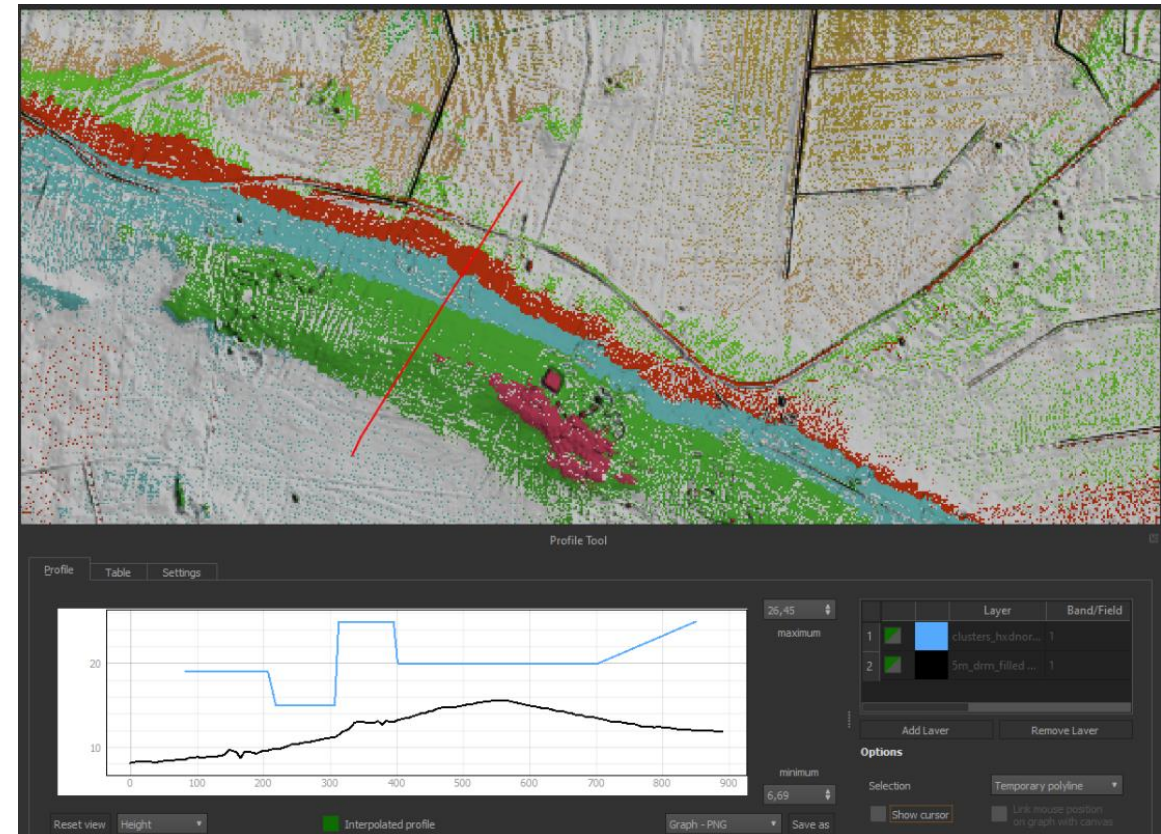
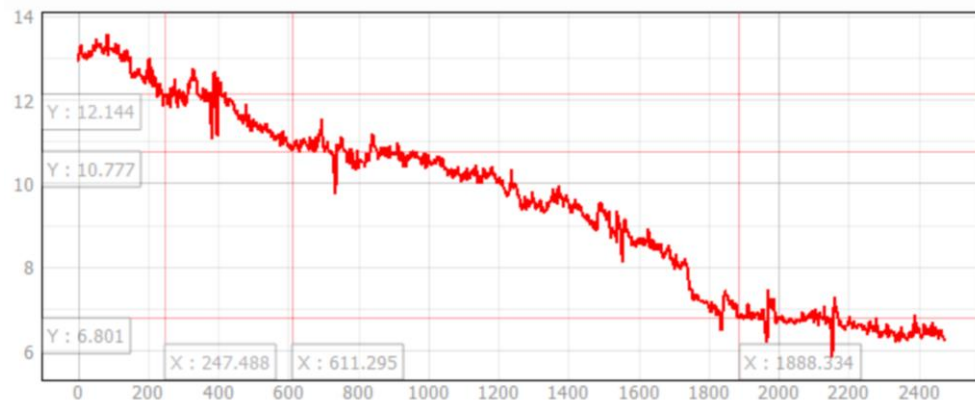


Seno krasta līniju modeļu izveide



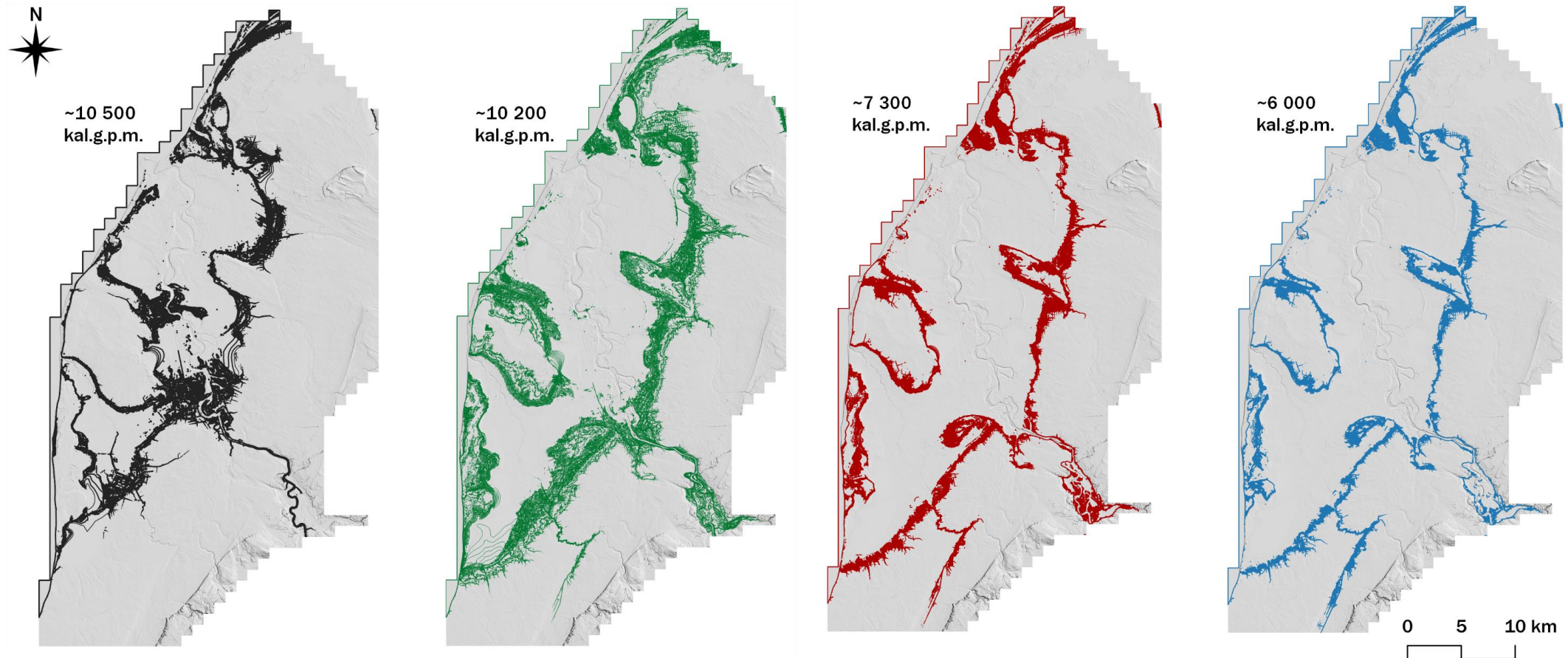
Modelēšanas rezultātu verifikācija

1. Atbilstība glacioizostatiskās pacelšanās virzienam un ātrumam
2. Saderība ar iepriekšējiem pētījumiem
3. Atbilstība reljefam (*QGIS* spraudnis *Profile Tool*)



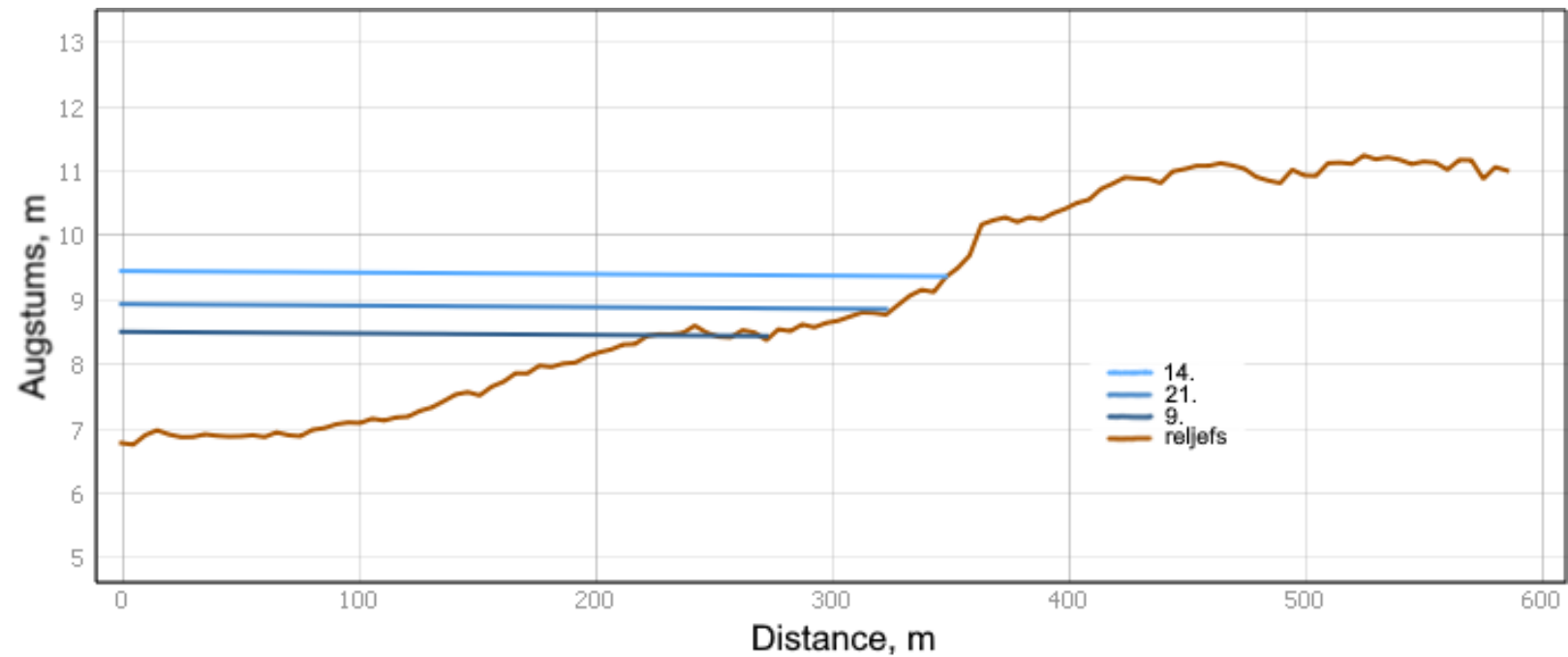
Senās Ventspils lagūnas piemērs

120 modeļi 4 laika posmiem \Rightarrow 25 verificēti modeļi



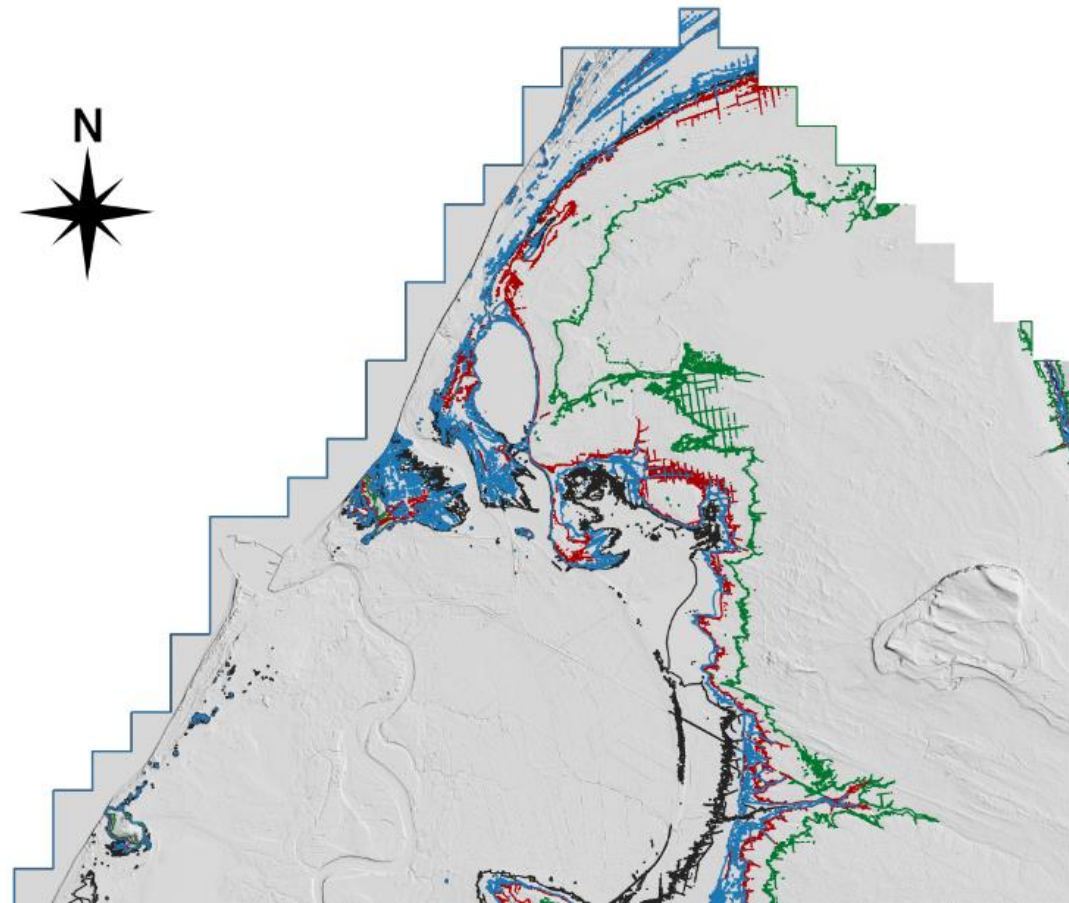
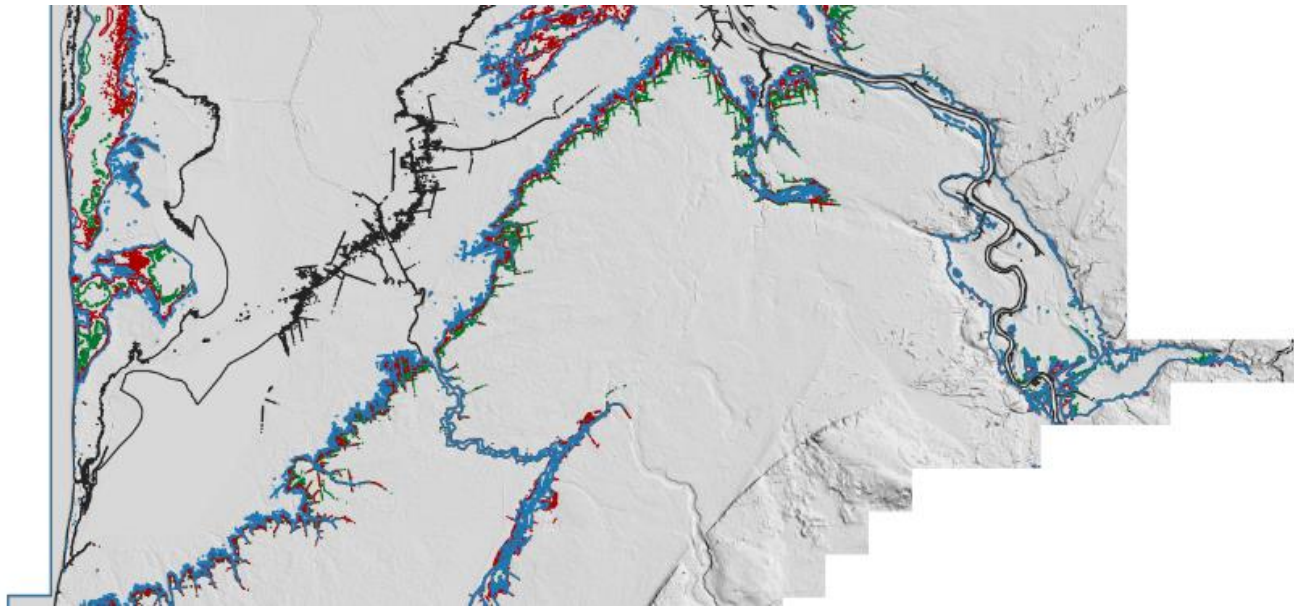
Senās Ventspils lagūnas piemērs

Krasta līnijas vai krasti?



Senās Ventspils lagūnas piemērs

Divas stadijas, tie paši krasti



Paldies!