WEBVTT

1 00:00:00.000 --> 00:00:02.613 <v Robert>Hi, I'm a Professor McDougal,</v>

2 00:00:06.358 --> 00:00:07.990 and Professor Wayne is also in the back.

3 00:00:07.990 --> 00:00:11.060 If you haven't signed in, please make sure that you pass

4 00:00:11.060 --> 00:00:13.310 this, get a chance to sign the sign in sheet.

5 00:00:14.590 --> 00:00:19.260 So today we are very, very privileged to be joined

6 00:00:19.260 --> 00:00:20.810 by Professor Naim Rashid

7 00:00:22.030 --> 00:00:25.360 from the University of North Carolina Chapel Hill,

8 00:00:25.360 --> 00:00:29.890 Professor Rashid got his bachelor's in biology from Duke,

9 00:00:29.890 --> 00:00:34.890 and his PhD in biostatistics from UNC Chapel Hill.

10 00:00:34.930 --> 00:00:39.930 He's the author of 34 publications, and he holds a patent

11 00:00:39.960 --> 00:00:44.410 on methods in composition for prognostic

12 00:00:44.410 --> 00:00:47.313 and/or diagnostic supply chain of pancreatic cancer.

13 00:00:48.170 --> 00:00:50.640 He's currently an associate professor at UNC Chapel Hill's

14 00:00:50.640 --> 00:00:53.710 department of biostatistics, and he's also affiliated

15 00:00:53.710 --> 00:00:56.903 with their comprehensive cancer center there.

16 00:00:59.100 --> 00:01:04.100 With that, Professor Rashid, would you like to take it away?

17 00:01:04.440 --> 00:01:05.920 <v ->Sure.</v>

18 00:01:05.920 --> 00:01:08.470 It looks like it says host disabled screen sharing.

19 00:01:10.344 --> 00:01:12.301 (chuckling)

20 00:01:12.301 --> 00:01:13.760 <v Robert>All right, give me one second.</v>

21 00:01:13.760 --> 00:01:14.823 Thank you.

22 00:01:16.760 --> 00:01:17.883 I'm trying to do.

23 00:01:26.781 --> 00:01:29.198 (indistinct)

24 00:01:33.645 --> 00:01:35.901 Okay, you should be, you should be able to come on now.

25 00:01:35.901 --> 00:01:36.984 <v ->All right.</v>

26 00:01:38.584 --> 00:01:39.873 Can you guys see my screen?

27 00:01:43.650 --> 00:01:44.483 All right.

28 00:01:47.537 --> 00:01:48.637 Can you guys see this?

29 00:01:49.840 --> 00:01:50.913 <v Robert>There we go.</v>

30 00:01:52.062 --> 00:01:52.895 Perfect. Thank you.

31 00:01:52.895 --> 00:01:53.850 <v ->Okay, great.</v>

32 00:01:53.850 --> 00:01:56.501 So yes, thanks to the department for inviting me to speak

33 00:01:56.501 --> 00:02:00.483 today, and also thanks to Robert and Wayne for organizing.

34 00:02:01.460 --> 00:02:04.420 And today I'll be talking about issues regarding

35 00:02:04.420 --> 00:02:07.500 replicability in terms of clinical prediction models,

36 00:02:07.500 --> 00:02:11.830 specifically in the context of genomic prediction models,

37 00:02:11.830 --> 00:02:13.423 derived from clinical trials.

38 00:02:16.080 --> 00:02:17.870 So as an overview, we'll be talking first a little bit

39 00:02:17.870 --> 00:02:20.670 about the problems of replicability in general,

40 00:02:20.670 --> 00:02:24.300 in scientific research, and also about specific issues

41 00:02:24.300 --> 00:02:28.040 in genomics itself, and then I'll be moving on to talking

42 00:02:28.040 --> 00:02:31.070 about a method that we've proposed to assist

43 00:02:31.070 --> 00:02:34.380 with issues regarding data integration, and learning

44 00:02:34.380 --> 00:02:37.680 in this environment when you have a heterogeneous data sets.

45 00:02:37.680 --> 00:02:39.860 I'll talk a little bit about a case study

46 00:02:39.860 --> 00:02:42.901 where we apply these practices to subtyping

47 00:02:42.901 --> 00:02:44.670 pancreatic cancer, touch on some current work

48 00:02:44.670 --> 00:02:46.581 that we're doing, and then end

49 00:02:46.581 --> 00:02:47.890 with some concluding thoughts.

50 00:02:47.890 --> 00:02:49.861 And feel free to interrupt, you know,

51 00:02:49.861 --> 00:02:52.211 as the talk is long, if you have any questions.

52 00:02:53.540 --> 00:02:55.650 So I'm now an associate professor in the department

53 00:02:55.650 --> 00:02:57.017 of biostatistics at UNC.

54 00:02:58.160 --> 00:03:00.430 My work generally involves problems

55 00:03:00.430 --> 00:03:04.730 surrounding cancer and genomics, and more recently

56 00:03:04.730 --> 00:03:07.390 we've been doing work regarding epigenomics.

57 00:03:07.390 --> 00:03:09.370 We just recently published a supply-connected package called

58 00:03:09.370 --> 00:03:13.120 Epigram for a consistence of differential key calling,

59 00:03:13.120 --> 00:03:15.480 and we've also done some work in model-based clustering.

60 00:03:15.480 --> 00:03:18.310 We published a package called, FSCSeq,

61 00:03:18.310 --> 00:03:21.780 which helps you derive and discover clusters

62 00:03:21.780 --> 00:03:23.830 from RNA seq data, while also determining

63 00:03:24.717 --> 00:03:25.550 clusters in specific genes.

64 00:03:25.550 --> 00:03:27.980 And today we'll be talking more about the topic

65 00:03:27.980 --> 00:03:30.340 of multi-study replicability, which is the topic

66 00:03:30.340 --> 00:03:33.710 of a paper that we published a year or two ago,

67 00:03:33.710 --> 00:03:36.570 and in our package that we've developed more recently,

68 00:03:36.570 --> 00:03:38.463 implementing some of the methods.

69 00:03:40.090 --> 00:03:42.660 So before I get deeper into the talk, one of the things

70 00:03:42.660 --> 00:03:45.130 I wanted to establish is this definition

71 00:03:45.130 --> 00:03:47.090 of what we mean by replicability.

72 00:03:47.090 --> 00:03:49.670 You might've heard the term reproducibility as well,

73 00:03:49.670 --> 00:03:52.430 and to make the distinction between the two terms,

74 00:03:52.430 --> 00:03:54.140 I'd like to define reproducibility in a way

75 00:03:54.140 --> 00:03:56.910 that Jeff Leak has defined in the past,

76 00:03:56.910 --> 00:03:59.410 where reproducibility is the ability to take

77 00:03:59.410 --> 00:04:02.540 coding data from a publication, and to rerun the code,

78 00:04:02.540 --> 00:04:05.630 and get the same results as the original publication.

79 00:04:05.630 --> 00:04:08.650 Where replicability, we're defining as the ability to be run

80 00:04:08.650 --> 00:04:10.980 an experiment generating new data, and get results

81 00:04:10.980 --> 00:04:12.780 that are quote, unquote ”consistent”

82 00:04:14.088 --> 00:04:15.560 with that of the original study.

83 00:04:15.560 --> 00:04:18.720 So in this sort of context, when it comes to replicability,

84 00:04:18.720 --> 00:04:21.890 you might've heard about publications that have come out

85 00:04:21.890 --> 00:04:23.773 in the past that talk about how there are issues

86 00:04:23.773 --> 00:04:27.600 regarding replicating the research that's been published

87 00:04:27.600 --> 00:04:29.570 in the scientific literature.

88 00:04:29.570 --> 00:04:32.280 This one paper in PLOS Medicine was published

89 00:04:32.280 --> 00:04:36.150 by, and that is in 2005, and there's been a number

90 00:04:36.150 --> 00:04:37.920 of publications that have come out since,

91 00:04:37.920 --> 00:04:40.880 talking about problems regarding replicability,

92 00:04:40.880 --> 00:04:43.290 and ways that we could potentially address it.

93 00:04:43.290 --> 00:04:45.820 And the problem has become large enough where it has

94 00:04:45.820 --> 00:04:48.840 its own Wikipedia entry talking about the crisis,

95 00:04:48.840 --> 00:04:51.300 and has a long list of examples that talks

96 00:04:51.300 --> 00:04:54.170 about issues regarding replicating results

97 00:04:54.170 --> 00:04:55.400 from the scientific studies.

98 00:04:55.400 --> 00:04:57.550 So this is something that has been a known issue

99 00:04:57.550 --> 00:05:00.320 for a while, and these problems also extend

100 00:05:00.320 --> 00:05:03.270 to situations where you want to, for example,

101 00:05:03.270 --> 00:05:06.300 develop clinical prediction models in genomics.

102 00:05:06.300 --> 00:05:10.280 So to give an example of this, let's say that we wanted to,

103 00:05:10.280 --> 00:05:13.200 in the population of metastatic breast cancer patients,

104 00:05:13.200 --> 00:05:15.710 we wanted to develop a model that predicts

105 00:05:15.710 --> 00:05:18.170 some clinical outcome Y, given a set

106 00:05:18.170 --> 00:05:20.530 of gene expression values X.

107 00:05:20.530 --> 00:05:23.020 And so the purpose of this sort of exercise is

108 00:05:23.020 --> 00:05:26.120 to hopefully translate this sort of model

109 00:05:26.120 --> 00:05:27.930 that we've developed, and apply it to the clinic,

110 00:05:27.930 --> 00:05:31.030 where we can use it for clinical decision-making.

111 00:05:31.030 --> 00:05:34.653 Now, if we have data from one particular trial

112 00:05:34.653 --> 00:05:36.960 that pertains to this patient population,

113 00:05:36.960 --> 00:05:39.020 and the same clinical outcome being measured,

114 00:05:39.020 --> 00:05:40.640 in addition to having gene expression data,

115 00:05:40.640 --> 00:05:42.640 let's say that we derived a model, let's say

116 00:05:42.640 --> 00:05:44.470 that we're modeling some sort of binary outcome,

117 00:05:44.470 --> 00:05:45.800 let's say tumor response.

118 00:05:45.800 --> 00:05:48.190 And in this model, we used a cost report,

119 00:05:48.190 --> 00:05:51.110 or penalized logistic regression model

120 00:05:51.110 --> 00:05:54.060 that we fit to the data to try and predict the outcome,

121 00:05:54.060 --> 00:05:55.940 given the gene expression values.

122 00:05:55.940 --> 00:05:58.770 And here we obtained, let's say, 12 genes

123 00:05:58.770 --> 00:06:03.640 after the fitting process, and the internal model 1 UNC

124 00:06:03.640 --> 00:06:05.733 on the sort of training subjects is 0.9.

125 00:06:06.740 --> 00:06:08.500 But then let's say there's another group at Duke

126 00:06:08.500 --> 00:06:10.870 that's using data from their clinical trial,

127 00:06:10.870 --> 00:06:13.197 and they have a larger sample size.

128 00:06:13.197 --> 00:06:15.870 They also found more genes, 65 genes,

129 00:06:15.870 --> 00:06:18.211 but have a slightly lower training at UNC.

130 00:06:18.211 --> 00:06:21.910 However, we really need to use external validation

131 00:06:21.910 --> 00:06:25.150 to sort of get an independent assessment of how well

132 00:06:25.150 --> 00:06:27.340 each one of these alternative models are doing.

133 00:06:27.340 --> 00:06:29.807 So let's say we have data from a similar study from Harvard,

134 00:06:29.807 --> 00:06:31.740 and we applied both these train models

135 00:06:32.615 --> 00:06:35.260 to the genomic data from that study at Harvard.

136 00:06:35.260 --> 00:06:37.790 We have the outcome information for those patients as well,

137 00:06:37.790 --> 00:06:42.153 so we can calculate how well the model predicts

138 00:06:42.153 --> 00:06:44.487 on those validation subjects.

139 00:06:44.487 --> 00:06:46.240 And we find here in this data set,

140 00:06:46.240 --> 00:06:48.740 model 2 seems to be doing better than model 1,

141 00:06:48.740 --> 00:06:50.870 but if you try this again with another data set

142 00:06:50.870 --> 00:06:53.470 from Michigan, you might find that model 1 is doing

143 00:06:53.470 --> 00:06:54.730 better, better than model 2.

144 00:06:54.730 --> 00:06:57.640 So the problem here is where we have researchers

145 00:06:57.640 --> 00:06:58.960 that are pointing fingers at each other,

146 00:06:58.960 --> 00:07:01.470 and it’s really hard to know, ”Well, who’s who’s right?”

147 00:07:01.470 --> 00:07:03.580 And why is this even happening in the first place,

148 00:07:03.580 --> 00:07:05.938 in terms of why do we get different genes, numbers of genes,

149 00:07:05.938 --> 00:07:08.797 and each of the models derived from study 1 and study 2?

150 00:07:08.797 --> 00:07:11.770 And why are we seeing very low performance

151 00:07:11.770 --> 00:07:13.620 in some of these validation datasets?

152 00:07:15.290 --> 00:07:17.330 So here’s an example from 2014,

153 00:07:17.330 --> 00:07:19.600 in the context of ovarian cancer.

154 00:07:19.600 --> 00:07:22.410 The authors basically collected 10 studies,

155 00:07:22.410 --> 00:07:24.063 all were microarray studies.

156 00:07:24.920 --> 00:07:27.200 The goal here was to predict overall survival

157 00:07:27.200 --> 00:07:29.550 in this population of ovarian cancer patients,

158 00:07:29.550 --> 00:07:31.870 given gene expression measurements

159 00:07:31.870 --> 00:07:33.800 from this microarray platform.

160 00:07:33.800 --> 00:07:34.633 So through a series

161 00:07:34.633 --> 00:07:38.640 of really complicated cross-fertilization approaches,

162 00:07:38.640 --> 00:07:40.430 the data was normalized, and harmonized

163 00:07:40.430 --> 00:07:43.413 across the studies, using a combination of ComBat

164 00:07:43.413 --> 00:07:45.639 and frozen RNA, and then they took

165 00:07:45.639 --> 00:07:47.640 14 published prediction models in the literature,

166 00:07:47.640 --> 00:07:50.970 and they applied each of those models to each

167 00:07:50.970 --> 00:07:53.255 of the subjects from these 10 studies, and they compared

168 00:07:53.255 --> 00:07:57.590 the model predictions across each subject.

169 00:07:57.590 --> 00:08:00.490 So each column here in this matrix is a patient,

170 00:08:00.490 --> 00:08:03.060 and each row is a different prediction model,

171 00:08:03.060 --> 00:08:06.260 and each cell represents the prediction

172 00:08:06.260 --> 00:08:08.090 from that model on that patient.

173 00:08:08.090 --> 00:08:11.700 So an ideal scenario, where we have the models generalizing

174 00:08:11.700 --> 00:08:14.480 and replicating across each of these individuals,

175 00:08:14.480 --> 00:08:15.860 we would expect to see the column,

176 00:08:15.860 --> 00:08:18.919 each column here to have the same color value,

177 00:08:18.919 --> 00:08:20.080 meaning that the predictions are consistent.

178 00:08:20.080 --> 00:08:22.220 But clearly we see here that the predictions are

179 00:08:22.220 --> 00:08:24.310 actually very inconsistent,

180 00:08:24.310 --> 00:08:26.060 and very different from each other.

181 00:08:27.230 --> 00:08:28.220 In addition, if you look

182 00:08:28.220 --> 00:08:30.410 at the individual risk prediction models

183 00:08:30.410 --> 00:08:31.990 that the authors used, there was also

184 00:08:31.990 --> 00:08:33.770 substantial differences in the genes

185 00:08:33.770 --> 00:08:36.210 that were selected in each of these models.

186 00:08:36.210 --> 00:08:39.760 So there's a max 2% overlap in terms of common genes

187 00:08:39.760 --> 00:08:41.350 between each of these approaches.

188 00:08:41.350 --> 00:08:43.150 And one thing to mention here is that each one

189 00:08:43.150 --> 00:08:45.380 of these risk-prediction models were derived

190 00:08:45.380 --> 00:08:48.270 from separate individual studies.

191 00:08:48.270 --> 00:08:50.631 So the question here is, you know, how exactly,

192 00:08:50.631 --> 00:08:53.669 if you were a clinician, you're eager to sort of take

193 00:08:53.669 --> 00:08:57.020 the results that you're seeing here,

194 00:08:57.020 --> 00:08:58.430 and extend to the clinic,

195 00:08:58.430 --> 00:09:00.860 which model do you use, which is right?

196 00:09:00.860 --> 00:09:02.610 Why are you seeing this level of variability?

197 00:09:02.610 --> 00:09:05.840 This is, of course, concerning, if you, if your goal is

198 00:09:05.840 --> 00:09:08.070 to move things towards the clinic, and this also has

199 00:09:08.070 --> 00:09:11.250 implications in terms of, you know, getting in the way

200 00:09:11.250 --> 00:09:12.980 of trying to approve the use of some

201 00:09:12.980 --> 00:09:15.453 of these, and for clinical use.

202 00:09:17.360 --> 00:09:18.950 So why is this happening?

203 00:09:18.950 --> 00:09:21.600 So there's been a lot of studies have been done

204 00:09:21.600 --> 00:09:24.487 that have tied issues to, obviously, sample size

205 00:09:24.487 --> 00:09:27.160 in the training studies, smaller sample sizes,

206 00:09:27.160 --> 00:09:30.710 and models trained on them may lead to more unstable models,

207 00:09:30.710 --> 00:09:32.182 or less accurate models.

208 00:09:32.182 --> 00:09:34.765 Between different studies, you might have

209 00:09:34.765 --> 00:09:36.080 different prevalences of the clinical outcome.

210 00:09:36.080 --> 00:09:38.640 In some studies, you might have higher levels of response,

211 00:09:38.640 --> 00:09:40.390 and other studies, you might have lower levels of response,

212 00:09:40.390 --> 00:09:42.920 for example, if you have this binary clinical outcome,

213 00:09:42.920 --> 00:09:46.290 and also there's issues regarding differences

214 00:09:46.290 --> 00:09:49.090 in lab conditions, where the genomic data was extracted.

215 00:09:49.090 --> 00:09:51.630 We've seen at Lineberger that, depending on the type

216 00:09:51.630 --> 00:09:54.570 of extraction, RNA extraction kit that you use,

217 00:09:54.570 --> 00:09:57.740 you might see differences in the expression of a gene,

218 00:09:57.740 --> 00:10:00.010 even from the same original tumor.

219 00:10:00.010 --> 00:10:01.640 And also the issue of batch placement,

220 00:10:01.640 --> 00:10:03.730 which has been widely talked about in the literature,

221 00:10:03.730 --> 00:10:06.170 where depending on the day you run the experiment,

222 00:10:06.170 --> 00:10:10.500 or the technician who's handling the data,

223 00:10:10.500 --> 00:10:12.023 you might see slight differences,

224 00:10:12.023 --> 00:10:14.263 technical differences in expression.

225 00:10:15.380 --> 00:10:16.810 There's also differences due to protocols.

226 00:10:16.810 --> 00:10:18.460 Some trials might have different inclusion

227 00:10:18.460 --> 00:10:20.560 and exclusion criteria, so they might be recruiting

228 00:10:20.560 --> 00:10:22.280 a slightly different patient population,

229 00:10:22.280 --> 00:10:23.640 even though they might be all

230 00:10:23.640 --> 00:10:25.240 in the context of metastatic breast cancer.

231 00:10:25.240 --> 00:10:29.161 All of these things can help impart heterogeneity

232 00:10:29.161 --> 00:10:33.590 between what the genomic data and the outcome data

233 00:10:33.590 --> 00:10:36.120 across different studies.

234 00:10:36.120 --> 00:10:38.710 In the context of genomic data in particular,

235 00:10:38.710 --> 00:10:41.280 there's also this aspect of data preprocessing.

236 00:10:41.280 --> 00:10:44.510 For the normalization taking that you use is very important,

237 00:10:44.510 --> 00:10:46.630 and we'll talk about that in a little bit.

238 00:10:46.630 --> 00:10:48.330 And it's a very critical part when it comes

239 00:10:48.330 --> 00:10:51.680 to training models, and trying to validate your model

240 00:10:51.680 --> 00:10:54.023 on other datasets, and depending on the type

241 00:10:54.023 --> 00:10:57.923 of normalization you use, this could also impact

242 00:10:57.923 --> 00:10:59.623 how well your model works.

243 00:11:00.480 --> 00:11:03.427 In addition, there's also differences in the potential way

244 00:11:03.427 --> 00:11:04.470 in which you measure gene expression.

245 00:11:04.470 --> 00:11:07.410 Some trials might use an older technology called microarray.

246 00:11:07.410 --> 00:11:08.940 I know other trials might use something

247 00:11:08.940 --> 00:11:11.490 relatively more recent called RNAC,

248 00:11:11.490 --> 00:11:12.593 or a particular trial might use

249 00:11:12.593 --> 00:11:14.910 a more targeted platform like NanoString.

250 00:11:14.910 --> 00:11:19.087 So the differences in platform also can lead to differences

251 00:11:19.087 --> 00:11:21.470 in your ability to help validate some of these studies.

252 00:11:21.470 --> 00:11:23.870 If you train something in marker rate, it's very difficult

253 00:11:23.870 --> 00:11:26.360 to take that model, and apply it to RNAC,

254 00:11:26.360 --> 00:11:29.900 because the expression values are just are just different.

255 00:11:29.900 --> 00:11:32.450 And so, as I mentioned before, this also impacts

256 00:11:32.450 --> 00:11:37.180 through to normalization on model performance as well.

257 00:11:37.180 --> 00:11:39.660 So the main thing to remember here is that

258 00:11:39.660 --> 00:11:43.080 the traditional way in which prediction models,

259 00:11:43.080 --> 00:11:46.130 based on genomic data for using the clinical training is

260 00:11:46.130 --> 00:11:49.343 typically on the results from a single study.

261 00:11:51.760 --> 00:11:53.510 To talk a little bit more about question

262 00:11:53.510 --> 00:11:57.260 of between-study normalization, and the purpose of this is

263 00:11:57.260 --> 00:12:00.360 to put the expression data on basically an even scale,

264 00:12:00.360 --> 00:12:02.330 which helps facilitate training.

265 00:12:02.330 --> 00:12:05.510 If there's global shifts, and some of the expression values

266 00:12:05.510 --> 00:12:08.820 in one sample versus another, it's very difficult to train

267 00:12:08.820 --> 00:12:11.090 an accurate model in that particular scenario.

268 00:12:11.090 --> 00:12:13.213 So normalization helps to align

269 00:12:13.213 --> 00:12:15.600 the expression you get from different samples,

270 00:12:15.600 --> 00:12:19.020 and hopefully across the between difference as well.

271 00:12:19.020 --> 00:12:23.090 And so the goal here is to eventually predict this outcome

272 00:12:23.090 --> 00:12:25.110 in a new patient, you plug in the genomic data

273 00:12:25.110 --> 00:12:28.190 from a new patient in order to get the predicted outcome

274 00:12:28.190 --> 00:12:30.350 for that patient based on that training model.

275 00:12:30.350 --> 00:12:33.650 So the, in order to do that, you also have to normalize

276 00:12:33.650 --> 00:12:35.910 the new data to the training data, right?

277 00:12:35.910 --> 00:12:38.151 Because you also want to put the new data on the same scale

278 00:12:38.151 --> 00:12:41.450 as a training data, and in the ideal scenario,

279 00:12:41.450 --> 00:12:43.610 you would want to make sure that the training samples

280 00:12:43.610 --> 00:12:47.150 that you use to train your original model are untouched,

281 00:12:47.150 --> 00:12:49.120 because what some people try to do is they try

282 00:12:49.120 --> 00:12:52.140 to sort of sidestep this normalization issue,

283 00:12:52.140 --> 00:12:54.644 they would combine the new data with the old training data,

284 00:12:54.644 --> 00:12:57.160 and renormalize everything at once.

285 00:12:57.160 --> 00:12:58.790 And the problem with this is that this changes

286 00:12:58.790 --> 00:13:00.727 your training sample values, and in a sense,

287 00:13:00.727 --> 00:13:03.640 would necessitate the fact that you need to retrain

288 00:13:03.640 --> 00:13:04.473 your old model again.

289 00:13:04.473 --> 00:13:06.950 And this leads to instability, and lack of stability

290 00:13:06.950 --> 00:13:09.333 over time in terms of the model itself.

291 00:13:10.270 --> 00:13:12.231 So in the prior example from ovarian cancer,

292 00:13:12.231 --> 00:13:14.950 this is not as big of an issue, because you have

293 00:13:14.950 --> 00:13:17.590 all the data you want to work with in hand.

294 00:13:17.590 --> 00:13:19.670 This is a retrospective study, you have 10 data sets,

295 00:13:19.670 --> 00:13:22.450 so you just normalize everything at the same time,

296 00:13:22.450 --> 00:13:23.960 that's in ComBat and frozen RNA.

297 00:13:23.960 --> 00:13:26.950 And so you can split up those studies into separate training

298 00:13:26.950 --> 00:13:30.750 and test studies, and they're all rated on the same scale.

299 00:13:30.750 --> 00:13:34.250 But the problem is that in practice, you're trying to do

300 00:13:34.250 --> 00:13:37.130 a prospective type of analysis, where when you train

301 00:13:37.130 --> 00:13:40.300 your model, you're normalizing all of the available studies

302 00:13:40.300 --> 00:13:43.690 you have, let's say, and then you use that to predict

303 00:13:43.690 --> 00:13:47.010 the outcome in a future patient, or a future study.

304 00:13:47.010 --> 00:13:51.150 And so the problem with that is that you have to find

305 00:13:51.150 --> 00:13:54.610 a good way to align, as I mentioned before,

306 00:13:54.610 --> 00:13:56.780 the data from that future study for your training samples,

307 00:13:56.780 --> 00:14:00.080 and that may not be an easy task to do,

308 00:14:00.080 --> 00:14:02.730 especially for some of the newer platforms like RNAC.

309 00:14:04.160 --> 00:14:06.165 So taking this problem a step further,

310 00:14:06.165 --> 00:14:09.830 what if there's no good cross study normalization approach

311 00:14:09.830 --> 00:14:12.200 that's available to begin with?

312 00:14:12.200 --> 00:14:15.200 This really is going to make things difficult in terms

313 00:14:15.200 --> 00:14:17.560 of the training in the model in the first place.

314 00:14:17.560 --> 00:14:20.860 Another more complicated problem is that you might have

315 00:14:20.860 --> 00:14:23.770 different types of platforms at that training time.

316 00:14:23.770 --> 00:14:26.040 For example, you might have the only type of data

317 00:14:26.040 --> 00:14:29.160 that’s available from one study is NanoString in one case,

318 00:14:29.160 --> 00:14:32.640 and another study it’s only RNAC, so what do you do?

319 00:14:32.640 --> 00:14:35.250 And looking forward, as platforms change,

320 00:14:35.250 --> 00:14:36.382 as technology evolves, you have different ways

321 00:14:36.382 --> 00:14:41.382 of measuring gene expression, for example.

322 00:14:41.950 --> 00:14:44.440 So what do you do with the models that are trained

323 00:14:44.440 --> 00:14:48.060 on old data, because you can’t apply them to the new data?

324 00:14:48.060 --> 00:14:49.770 So oftentimes you find this situation

325 00:14:49.770 --> 00:14:53.470 where you have to retrain new models on these new platforms,

326 00:14:53.470 --> 00:14:57.000 and the old models are not able to be applied

327 00:14:57.000 --> 00:14:58.440 directly to this new data types.

328 00:14:58.440 --> 00:15:00.690 So that leads to waste here.

329 00:15:00.690 --> 00:15:03.370 So if you take all of these problems together,

330 00:15:03.370 --> 00:15:07.320 regarding cross-study normalization,

331 00:15:07.320 --> 00:15:09.300 and changes in platform,

332 00:15:09.300 --> 00:15:11.390 and a lot of the other issues, you know,

333 00:15:11.390 --> 00:15:13.280 regarding replicability that I mentioned,

334 00:15:13.280 --> 00:15:16.580 it’s no wonder that there’s only a small handful

335 00:15:16.580 --> 00:15:21.430 of expression-based clinically applicable assets have been

336 00:15:21.430 --> 00:15:23.777 approved by the FDA, like Oncotype DX, MammaPrint,

337 00:15:23.777 --> 00:15:27.203 and Prosigna, because this is a very, very tough problem.

338 00:15:29.884 --> 00:15:32.600 So I want to move on with that, to an approach

339 00:15:32.600 --> 00:15:36.130 that we proposed to help tackle this sort of issue

340 00:15:36.130 --> 00:15:39.210 by using this idea of multi-study learning,

341 00:15:39.210 --> 00:15:43.020 where instead of just using, and deriving, and generating

342 00:15:43.020 --> 00:15:44.810 models from individual studies, we combine data

343 00:15:44.810 --> 00:15:47.790 from multiple studies together, and create a consensus model

344 00:15:47.790 --> 00:15:50.110 that we use for prediction, which will hopefully be

345 00:15:50.110 --> 00:15:54.140 more stable, and more accurate down the road.

346 00:15:54.140 --> 00:15:56.400 So this approach of combining data is called

347 00:15:56.400 --> 00:15:59.190 horizontal data integration, where we're merging data

348 00:15:59.190 --> 00:16:01.360 from let's say K different studies.

349 00:16:01.360 --> 00:16:04.300 And the pro of this approach is that we get increased power,

350 00:16:04.300 --> 00:16:06.160 and the ability to reach some sort of consensus

351 00:16:06.160 --> 00:16:08.860 across these different studies.

352 00:16:08.860 --> 00:16:11.650 The negative is that the effect of a gene

353 00:16:11.650 --> 00:16:13.710 and its relationship to outcome may actually vary

354 00:16:13.710 --> 00:16:16.040 across studies, and also by, you know, depending on,

355 00:16:16.040 --> 00:16:18.940 and also the way that you normalize the genes may also vary

356 00:16:18.940 --> 00:16:21.178 across studies too if we're using published data

357 00:16:21.178 --> 00:16:23.630 from some prior publication.

358 00:16:23.630 --> 00:16:25.470 There's also this issue of sample size and balance.

359 00:16:25.470 --> 00:16:27.630 You might have a study that has 500 subjects,

360 00:16:27.630 --> 00:16:29.860 and another one that might have 200 subjects.

361 00:16:29.860 --> 00:16:33.820 So there are some methods that were designed to account for

362 00:16:33.820 --> 00:16:36.190 between-study heterogeneity after you do

363 00:16:36.190 --> 00:16:37.830 horizontal data integration.

364 00:16:37.830 --> 00:16:41.040 One is called the meta-lasso, another is called

365 00:16:41.040 --> 00:16:43.590 the AW statistic, but these two methods don't really have

366 00:16:43.590 --> 00:16:46.370 any prediction aspect about them.

367 00:16:46.370 --> 00:16:48.496 They're more about feature selection.

368 00:16:48.496 --> 00:16:50.420 Ensembling is one approach that can directly account

369 00:16:50.420 --> 00:16:52.310 for between-study heterogeneity

370 00:16:52.310 --> 00:16:54.350 after horizontal data integration, but there's

371 00:16:54.350 --> 00:16:56.870 no explicit future selection step here.

372 00:16:56.870 --> 00:16:58.800 But all of these approaches assume

373 00:16:58.800 --> 00:17:01.670 that the data has been pre-normalized.

374 00:17:01.670 --> 00:17:03.350 As we talked about before,

375 00:17:03.350 --> 00:17:06.820 for prospective decision-making, based off a train model,

376 00:17:06.820 --> 00:17:10.070 that might be prohibitive in some cases,

377 00:17:10.070 --> 00:17:13.380 and we need a strategy also to easily predict

378 00:17:13.380 --> 00:17:17.153 and apply these models in new patients.

379 00:17:20.260 --> 00:17:24.080 Okay, so moving on, we're going to talk first

380 00:17:24.080 --> 00:17:26.670 about this issue of how do we integrate data,

381 00:17:26.670 --> 00:17:30.300 and sort of sidestep this normalization problem

382 00:17:30.300 --> 00:17:33.190 at training time, and also at test time where we,

383 00:17:33.190 --> 00:17:35.040 when we try to predict in new subjects?

384 00:17:35.040 --> 00:17:38.520 So the approach that we put forth is to use

385 00:17:38.520 --> 00:17:40.860 what's called top scoring pairs, which you can think of

386 00:17:40.860 --> 00:17:44.560 as a rank-based transformation of the original set

387 00:17:44.560 --> 00:17:47.320 of gene expression values from a patient.

388 00:17:47.320 --> 00:17:49.510 So the idea here originally,

389 00:17:49.510 --> 00:17:50.630 when top scoring pairs were introduced,

390 00:17:50.630 --> 00:17:53.390 was you're trying to find a pair of genes

391 00:17:53.390 --> 00:17:56.390 where it's such that if the expression of gene A

392 00:17:56.390 --> 00:17:58.908 in the pair is greater than gene B, that would imply

393 00:17:58.908 --> 00:18:02.970 that the, let's say, the subtype for that individual is,

394 00:18:02.970 --> 00:18:05.490 say, subtype one, and if it's less,

395 00:18:05.490 --> 00:18:09.080 then that implies subtype zero with high probability.

396 00:18:09.080 --> 00:18:11.760 Now, in this case, this sort of approach was developed

397 00:18:11.760 --> 00:18:14.100 with when one has a binary outcome variable

398 00:18:14.100 --> 00:18:15.070 that you care about.

399 00:18:15.070 --> 00:18:17.430 In this case, we're talking about subtype,

400 00:18:17.430 --> 00:18:20.040 but it could also be tumor response or something else.

401 00:18:20.040 --> 00:18:22.070 So essentially what you're doing is that you're taking

402 00:18:22.070 --> 00:18:25.270 these continuous measurements in terms of gene expression,

403 00:18:25.270 --> 00:18:30.270 or integer, and you are converting that, transforming

404 00:18:30.600 --> 00:18:32.230 that into basically a binary predictor,

405 00:18:32.230 --> 00:18:34.457 which takes on the value of the zero or one.

406 00:18:34.457 --> 00:18:38.210 And the hope is that that particular transformed value is

407 00:18:38.210 --> 00:18:41.300 going to be associated with this binary outcome.

408 00:18:41.300 --> 00:18:43.760 So the simple assumption in this scenario is

409 00:18:43.760 --> 00:18:46.100 that the relative rank of these genes

410 00:18:46.100 --> 00:18:50.810 in a given sample is predictive of subtype, and that's it.

411 00:18:50.810 --> 00:18:54.490 And so the example here I have on the right is an example

412 00:18:54.490 --> 00:18:57.790 of two genes, GSTP1 and ESR1.

413 00:18:57.790 --> 00:18:59.928 And so you can see here that if you're

414 00:18:59.928 --> 00:19:02.300 in the upper left quadrant, this is where this gene is

415 00:19:02.300 --> 00:19:04.860 greater than this gene expression, it's implying

416 00:19:04.860 --> 00:19:07.648 the triangle subtype with high probability,

417 00:19:07.648 --> 00:19:10.900 and otherwise it implies the circle subtype.

418 00:19:10.900 --> 00:19:14.350 So that's the general idea of what we're going for here.

419 00:19:14.350 --> 00:19:16.480 It's a sort of a rank-based transformation

420 00:19:16.480 --> 00:19:19.643 of the original continuous predictor space.

421 00:19:20.750 --> 00:19:22.100 So the nice thing about this approach,

422 00:19:22.100 --> 00:19:24.643 because we're only based on the simple assumption, right?

423 00:19:24.643 --> 00:19:26.710 That we're only caring about the relative rank

424 00:19:26.710 --> 00:19:28.880 within a subject, this makes

425 00:19:28.880 --> 00:19:32.450 this particular new transformed predictor

426 00:19:32.450 --> 00:19:35.710 relatively invariant to batch effects, pre-normalization,

427 00:19:35.710 --> 00:19:39.410 and it also most importantly, simplifies merging data

428 00:19:39.410 --> 00:19:40.580 from different studies.

429 00:19:40.580 --> 00:19:43.090 Everything is now on the same scale, zero to one,

430 00:19:43.090 --> 00:19:44.987 so it's very easy to paste together the data

431 00:19:44.987 --> 00:19:49.910 from different studies, and we can sidestep this problem

432 00:19:49.910 --> 00:19:52.870 of trying to pick a cross-normalization approach,

433 00:19:52.870 --> 00:19:55.803 and then work in this sort of transformed space.

434 00:19:56.840 --> 00:19:59.130 The other nice thing is that this is easily computable

435 00:19:59.130 --> 00:20:00.690 for new patients as well.

436 00:20:00.690 --> 00:20:02.670 If you have a new patient that comes into clinic,

437 00:20:02.670 --> 00:20:04.220 you just check to see whether the gene A is

438 00:20:04.220 --> 00:20:06.290 greater than gene B in terms of expression,

439 00:20:06.290 --> 00:20:11.290 and then you have your value for this top scoring pair,

440 00:20:11.350 --> 00:20:14.430 and we don't have to worry as much about normalizing

441 00:20:14.430 --> 00:20:17.740 this patient's raw gene spectrum data

442 00:20:17.740 --> 00:20:21.470 to the training sample expression values.

443 00:20:21.470 --> 00:20:23.360 So essentially what we're doing here is that we're,

444 00:20:23.360 --> 00:20:25.700 let's enumerate all possible gene pairs for us,

445 00:20:25.700 --> 00:20:28.200 instead of a candidate genes, and each column here

446 00:20:28.200 --> 00:20:30.530 in this matrix shown on the right pertains

447 00:20:30.530 --> 00:20:33.867 to the zero one values for a particular gene pair J.

448 00:20:33.867 --> 00:20:37.960 And so this value takes the value of one, it is greater

449 00:20:37.960 --> 00:20:41.200 than B, in sample I, in pair j, and zero otherwise.

450 00:20:41.200 --> 00:20:44.603 And then we merge over the common top scoring pairs.

451 00:20:46.070 --> 00:20:49.050 So in this example have data from four different studies,

452 00:20:49.050 --> 00:20:50.420 each indicator by a different color here

453 00:20:50.420 --> 00:20:53.750 in the first track, and this data pertains to data

454 00:20:53.750 --> 00:20:54.900 from two different platforms,

455 00:20:54.900 --> 00:20:56.437 and three different cancer types.

456 00:20:56.437 --> 00:20:59.220 And so the clinical outcome here is binary subtype,

457 00:20:59.220 --> 00:21:02.220 which is given by the orange and the blue color here.

458 00:21:02.220 --> 00:21:05.350 So you can see here that we enumerated the TSPs,

459 00:21:05.350 --> 00:21:07.190 we merged the data together, and now we have

460 00:21:07.190 --> 00:21:09.340 this transformed predictor agents.

461 00:21:09.340 --> 00:21:10.430 And the interesting thing is

462 00:21:10.430 --> 00:21:12.620 that you can definitely see some patterning here.

463 00:21:12.620 --> 00:21:15.290 With any study where you have a particular set of TSPs

464 00:21:15.290 --> 00:21:18.950 that had taken a value of one, when the subtype is blue,

465 00:21:18.950 --> 00:21:20.850 and it flips when it's orange.

466 00:21:20.850 --> 00:21:24.230 And we see the same general pattern seem to replicate

467 00:21:24.230 --> 00:21:25.380 across different studies,

468 00:21:25.380 --> 00:21:29.168 but not every top scoring pair changes the same way

469 00:21:29.168 --> 00:21:31.700 across different studies.

470 00:21:31.700 --> 00:21:34.970 So if we cluster the rows here, we can also see

471 00:21:34.970 --> 00:21:38.120 some patterns sort of persist where we see

472 00:21:38.120 --> 00:21:39.770 some clustering by subtype,

473 00:21:39.770 --> 00:21:41.830 but also some clustering by study as well.

474 00:21:41.830 --> 00:21:44.620 And so what this implies is that there's a relationship

475 00:21:44.620 --> 00:21:47.108 between TSPs and subtypes, and that can vary across studies,

476 00:21:47.108 --> 00:21:50.107 which is not too different from what we've talked

477 00:21:50.107 --> 00:21:51.380 about regarding the issues we've seen

478 00:21:51.380 --> 00:21:53.339 in replicability in the past.

479 00:21:53.339 --> 00:21:57.460 So ideally we would like to see a particular gene pair,

480 00:21:57.460 --> 00:22:00.810 or TSP vector here take on a value of one,

481 00:22:00.810 --> 00:22:02.500 only when there's the orange subtype,

482 00:22:02.500 --> 00:22:04.940 and zero in the blue subtype, or vice versa.

483 00:22:04.940 --> 00:22:06.670 And we wanted to see this pattern replicated

484 00:22:06.670 --> 00:22:09.680 across patients in studies, but we see obviously

485 00:22:09.680 --> 00:22:11.840 that that's not the case.

486 00:22:11.840 --> 00:22:14.650 So the question now that we've sort of introduced,

487 00:22:14.650 --> 00:22:16.530 or proposed is this sort of approach to simplify

488 00:22:16.530 --> 00:22:18.520 data merging in normalization.

489 00:22:18.520 --> 00:22:20.020 The question now that we're sort of dealing

490 00:22:20.020 --> 00:22:22.066 with is well, how do we actually now find

491 00:22:22.066 --> 00:22:25.830 features that are consistent across different studies

492 00:22:25.830 --> 00:22:28.560 in their relationship with outcome, and also estimate

493 00:22:28.560 --> 00:22:31.793 their study-level effect, and then use them for prediction?

494 00:22:32.860 --> 00:22:35.408 So that leads us to the second part of our paper,

495 00:22:35.408 --> 00:22:39.227 where we developed a model to help select

496 00:22:39.227 --> 00:22:42.027 these particular study-consistent features

497 00:22:42.027 --> 00:22:47.027 while accounting for study-level heterogeneity.

498 00:22:47.100 --> 00:22:49.410 So to sort of illustrate the idea behind this,

499 00:22:49.410 --> 00:22:51.700 let's just start with a simple simulation

500 00:22:51.700 --> 00:22:54.130 where we're not doing any normalization,

501 00:22:54.130 --> 00:22:56.310 we're not worrying about resuming, everything's fine

502 00:22:56.310 --> 00:22:58.730 in terms of the expression values,

503 00:22:58.730 --> 00:23:00.170 and we're not doing any selection,

504 00:23:00.170 --> 00:23:02.900 no TSP transmission either.

505 00:23:02.900 --> 00:23:04.760 So we're going to assimilate data pertaining

506 00:23:04.760 --> 00:23:06.380 to two, let's say, known biomarkers

507 00:23:06.380 --> 00:23:08.550 that are associated with binary subtype.

508 00:23:08.550 --> 00:23:10.607 We're going to generate K datasets,

509 00:23:10.607 --> 00:23:12.200 and we're going to try three different strategies

510 00:23:12.200 --> 00:23:14.690 for learning a prediction model two to these data sets.

511 00:23:14.690 --> 00:23:18.070 And at the end, we're going to validate each of those models

512 00:23:18.070 --> 00:23:18.903 on an externally-generated data set

513 00:23:18.903 --> 00:23:21.610 to compare their prediction performance.

514 00:23:21.610 --> 00:23:25.390 So to do this, we're going to fit and assume for each study

515 00:23:25.390 --> 00:23:27.790 that we can fit it with a logistic regression model

516 00:23:27.790 --> 00:23:30.640 to model by our outcome with these two predictors,

517 00:23:30.640 --> 00:23:32.410 and in generating these K data sets,

518 00:23:32.410 --> 00:23:34.940 we're going to vary the number of with respect to K.

519 00:23:34.940 --> 00:23:37.690 So we might generate two trained data sets five or 10,

520 00:23:37.690 --> 00:23:39.770 and also change the total sample size of each one,

521 00:23:39.770 --> 00:23:41.830 and make sure that the sample sizes are in balanced

522 00:23:41.830 --> 00:23:44.790 across the different studies, and then assume

523 00:23:44.790 --> 00:23:49.510 values for the coefficients for each of these predictors

524 00:23:49.510 --> 00:23:52.750 to be these values here, and lastly, to induce some sort

525 00:23:52.750 --> 00:23:55.787 of heterogeneity across the different training datasets,

526 00:23:55.787 --> 00:23:59.410 we're gonna add in sort of like a random value drop

527 00:23:59.410 --> 00:24:01.910 from the normal distribution, where we're assuming

528 00:24:02.786 --> 00:24:04.610 this level of variance for this value.

529 00:24:04.610 --> 00:24:06.660 So basically we're just injecting heterogeneity

530 00:24:06.660 --> 00:24:08.403 into this data generation process.

531 00:24:09.310 --> 00:24:10.880 So after we generate the training studies,

532 00:24:10.880 --> 00:24:12.940 then we're going to apply three different ways

533 00:24:12.940 --> 00:24:15.370 or strategies to the training data.

534 00:24:15.370 --> 00:24:17.330 The first is the individual study approach,

535 00:24:17.330 --> 00:24:19.730 which we've talked about before, where you train

536 00:24:19.730 --> 00:24:22.390 a generalized model separately for each study.

537 00:24:22.390 --> 00:24:24.600 The second approach is where you merge the data.

538 00:24:24.600 --> 00:24:26.430 Again, we're ignoring the normalization problem here

539 00:24:26.430 --> 00:24:29.770 in simulation, obviously, and then train a single GLMM

540 00:24:29.770 --> 00:24:31.870 for the combined data, and then lastly,

541 00:24:31.870 --> 00:24:33.660 we're going to merge the data, and train

542 00:24:33.660 --> 00:24:35.120 a generalized linear mixed model,

543 00:24:35.120 --> 00:24:38.047 where we explicitly account for a random intercept,

544 00:24:38.047 --> 00:24:40.610 and a random slope for each predictor,

545 00:24:40.610 --> 00:24:44.500 assuming, you know, a study-level random effect.

546 00:24:44.500 --> 00:24:48.490 So after we do that, we'll generate a validation dataset

547 00:24:48.490 --> 00:24:52.224 from the same approach above, and then predict outcome

548 00:24:52.224 --> 00:24:54.500 in this validation dataset with respect

549 00:24:54.500 --> 00:24:57.400 to the models derived from each of these three strategies.

550 00:24:59.180 --> 00:25:01.460 So if we look at the individual strategy performance,

551 00:25:01.460 --> 00:25:03.820 where we fit a GLM logistical regression model

552 00:25:03.820 --> 00:25:06.010 separately for each study, and then apply it

553 00:25:06.010 --> 00:25:07.710 to this validation data set, we can check

554 00:25:07.710 --> 00:25:10.580 the prediction accuracy, we can find that,

555 00:25:10.580 --> 00:25:13.860 due to the induced level of heterogeneity

556 00:25:13.860 --> 00:25:15.800 between studies in predictor effects,

557 00:25:15.800 --> 00:25:18.060 in one study, we do really poorly,

558 00:25:18.060 --> 00:25:20.070 and another study we do really well,

559 00:25:20.070 --> 00:25:24.060 and this variation is entirely due to variations

560 00:25:24.060 --> 00:25:26.580 in the gene subtype relationship.

561 00:25:26.580 --> 00:25:28.830 And these predictions obviously vary as a result

562 00:25:28.830 --> 00:25:30.080 across the different studies.

563 00:25:30.080 --> 00:25:32.440 And this will reflect a little bit of what we see

564 00:25:32.440 --> 00:25:35.030 in some of the examples that we showed earlier,

565 00:25:35.030 --> 00:25:38.003 studies that were trained on different data sets.

566 00:25:40.410 --> 00:25:42.550 And then the second approach is where we combine

567 00:25:42.550 --> 00:25:45.560 the data sets, and train a single logistical question model

568 00:25:45.560 --> 00:25:46.430 to predict outcome.

569 00:25:46.430 --> 00:25:48.530 And so we see what the median prediction error is better

570 00:25:48.530 --> 00:25:51.630 than most of the models here, but if we fit the GLMM,

571 00:25:51.630 --> 00:25:53.640 the median prediction (indistinct) gets better

572 00:25:53.640 --> 00:25:55.800 than some of the other approaches here.

573 00:25:55.800 --> 00:25:57.890 So this is basically just one example.

574 00:25:57.890 --> 00:26:00.120 So we did this over and over a hundred times

575 00:26:00.120 --> 00:26:02.640 for every single possible simulation condition,

576 00:26:02.640 --> 00:26:07.130 varying K, and the heterogeneity across different studies.

577 00:26:07.130 --> 00:26:09.560 And some of the things that we found was that

578 00:26:09.560 --> 00:26:12.110 the individual study approach had, as you can see,

579 00:26:12.110 --> 00:26:14.460 the worst prediction error overall,

580 00:26:14.460 --> 00:26:16.610 combining the data improved this a little bit,

581 00:26:16.610 --> 00:26:20.720 but the estimates for the coefficients

582 00:26:20.720 --> 00:26:23.210 from the combined GLMM were still biased.

583 00:26:23.210 --> 00:26:26.720 There's supposed to be two in this extreme scenario.

584 00:26:26.720 --> 00:26:30.660 And a kind of heterogeneity with the GLMM mixed model had

585 00:26:30.660 --> 00:26:32.460 the best performance out of the rest,

586 00:26:32.460 --> 00:26:35.004 and also had the lowest bias in terms

587 00:26:35.004 --> 00:26:38.630 of the regression coefficients as well.

588 00:26:38.630 --> 00:26:42.150 So this is great, but we also have a lot

589 00:26:42.150 --> 00:26:43.888 of potential types of pairs.

590 00:26:43.888 --> 00:26:46.700 We can't really estimate them all

591 00:26:46.700 --> 00:26:49.800 with a GLMM mixed model, so we need to find a way

592 00:26:49.800 --> 00:26:52.030 where we can, at least in reasonable dimension,

593 00:26:52.030 --> 00:26:54.610 figure out a way which fixed effects are non-zero,

594 00:26:54.610 --> 00:26:56.100 while accounting for, you know,

595 00:26:56.100 --> 00:26:58.850 this sort of study-level heterogeneity for each effect.

596 00:27:00.460 --> 00:27:05.126 So this led us to develop a pGLMM, which is basically

597 00:27:05.126 --> 00:27:08.310 a high-dimensional generalized intermixed model,

598 00:27:08.310 --> 00:27:10.770 where we are able to select fixed and random effects

599 00:27:10.770 --> 00:27:13.420 simultaneously using a penalization framework.

600 00:27:13.420 --> 00:27:16.740 So essentially here, we're assuming that all the predictors

601 00:27:16.740 --> 00:27:18.740 in the model, we assume a random effect,

602 00:27:19.606 --> 00:27:23.046 a random slope for each one, and so we were aiming to select

603 00:27:23.046 --> 00:27:27.750 the features that have non-zero fixed effects

604 00:27:27.750 --> 00:27:29.540 in this particular approach, and indeed we're assuming

605 00:27:29.540 --> 00:27:31.550 these are going to be study-consistent.

606 00:27:31.550 --> 00:27:34.820 And to do this, we're going to reorganize

607 00:27:34.820 --> 00:27:38.040 the linear predictor from the standard GLMM,

608 00:27:38.040 --> 00:27:41.110 so basically we're starting with the same general likelihood

609 00:27:41.110 --> 00:27:44.220 for, you know, the generalized mixed model.

610 00:27:44.220 --> 00:27:49.024 Here, Y is our outcome, X is our predictor,

611 00:27:49.024 --> 00:27:53.040 alpha is the, alpha K is the random effect

612 00:27:53.040 --> 00:27:58.040 for the case study, fi here is typically assumed to be

613 00:27:58.150 --> 00:28:02.130 multi, very normal, means zero, and a covariant

614 00:28:02.130 --> 00:28:05.140 on some sort of unstructured covariance matrix typically.

615 00:28:05.140 --> 00:28:08.930 And so to sort of simplify this, we factor out

616 00:28:08.930 --> 00:28:10.390 the random effects covariance matrix,

617 00:28:10.390 --> 00:28:12.110 and incorporate into the linear predictor.

618 00:28:12.110 --> 00:28:15.950 And with some more reorganizing, now we're able to select

619 00:28:15.950 --> 00:28:20.950 the fixed effects and determine which random effects have

620 00:28:21.420 --> 00:28:23.600 true non-covariance, using this sort

621 00:28:23.600 --> 00:28:25.580 of joint penalization framework.

622 00:28:25.580 --> 00:28:27.540 If you want more detail, you can check out the publication

623 00:28:27.540 --> 00:28:31.340 that I linked above, and I also forgot to send out

624 00:28:31.340 --> 00:28:33.010 the link to this talk here.

625 00:28:33.010 --> 00:28:35.470 I'll do that right now, in case you want to check out

626 00:28:35.470 --> 00:28:38.283 some of the publications that I'm linking in this talk.

627 00:28:40.660 --> 00:28:42.330 Okay, so how do we do this estimation?

628 00:28:42.330 --> 00:28:44.270 And we use that penalized NCM algorithm,

629 00:28:44.270 --> 00:28:46.510 where in each step we're drawing from the posterior

630 00:28:46.510 --> 00:28:47.990 with respect to the random effects, given

631 00:28:47.990 --> 00:28:50.070 the current aspects of the parameters,

632 00:28:50.070 --> 00:28:55.070 and the observed data, using Metropolis point of Gibbs.

633 00:28:55.180 --> 00:28:58.262 In the R packets, I'm going to talk about in a little bit,

634 00:28:58.262 --> 00:29:03.000 we update this to using a Hamiltonian Monte Carlo,

635 00:29:03.000 --> 00:29:03.980 but in the original version,

636 00:29:03.980 --> 00:29:06.270 we use Metropolis point of Gibbs, where we skipped

637 00:29:07.120 --> 00:29:09.360 components that had zero variance from the M-STEP.

638 00:29:09.360 --> 00:29:11.938 And then we use, in the M-step,

639 00:29:11.938 --> 00:29:13.940 two conditional maximization steps

640 00:29:13.940 --> 00:29:17.110 where we first update data, given the draws

641 00:29:17.110 --> 00:29:20.200 from the E-step, and the prior estimates for gamma here,

642 00:29:20.200 --> 00:29:23.740 and then up to gamma using a group penalty.

643 00:29:23.740 --> 00:29:25.400 So we use a couple of other tricks

644 00:29:25.400 --> 00:29:27.060 to speed up performance here.

645 00:29:27.060 --> 00:29:28.530 I won't go too much into the details there,

646 00:29:28.530 --> 00:29:31.713 but you can check out the paper for more detail on that.

647 00:29:33.330 --> 00:29:34.570 But with this approach, one of the things

648 00:29:34.570 --> 00:29:36.579 that we were able to show was that we have

649 00:29:36.579 --> 00:29:39.290 similar conclusions regarding bias and prediction error,

650 00:29:39.290 --> 00:29:41.420 as in the simple setup we had before,

651 00:29:41.420 --> 00:29:43.390 where in this particular situation, we're simulating

652 00:29:43.390 --> 00:29:46.920 a bunch of predictors that do not have any association

653 00:29:46.920 --> 00:29:50.760 with outcome, either 10 to 50 extra predictors,

654 00:29:50.760 --> 00:29:53.410 or there's only two that are actually truly relevant.

655 00:29:54.480 --> 00:29:55.920 And so the prediction error in this model

656 00:29:55.920 --> 00:29:58.650 after this penalized selection process is

657 00:29:58.650 --> 00:30:01.320 generally the same, if not a little bit worse.

658 00:30:01.320 --> 00:30:03.440 And one thing that we find here is that

659 00:30:03.440 --> 00:30:04.940 the parameters are selected

660 00:30:05.782 --> 00:30:07.570 by the individual study approach we're applying now

661 00:30:07.570 --> 00:30:09.960 at penalized distribution regression model has

662 00:30:09.960 --> 00:30:12.859 a low sensitivity to detect the true predictors,

663 00:30:12.859 --> 00:30:15.542 and a higher false positive rate in terms of selecting

664 00:30:15.542 --> 00:30:17.210 predictors that aren't associated

665 00:30:17.210 --> 00:30:18.880 with outcome and simulation.

666 00:30:18.880 --> 00:30:22.660 And what we find here also is that the approach

667 00:30:22.660 --> 00:30:26.050 that we developed had a much better sensitivity

668 00:30:26.050 --> 00:30:27.800 compared to other approaches for selecting

669 00:30:27.800 --> 00:30:29.850 the true predictors when accounting

670 00:30:29.850 --> 00:30:31.723 for study-level homogeneity,

671 00:30:31.723 --> 00:30:33.183 and the lower false positive rate as well.

672 00:30:36.060 --> 00:30:39.080 The example data sets that I talked about before,

673 00:30:39.080 --> 00:30:43.160 the four ones that I showed a figure up earlier,

674 00:30:43.160 --> 00:30:45.030 we did a whole data study analysis where we trained

675 00:30:45.030 --> 00:30:48.110 on three studies and held out one of the studies.

676 00:30:48.110 --> 00:30:50.970 We found that, you know, the approach that we put forward

677 00:30:50.970 --> 00:30:53.730 that put combining the data using our TSP approach,

678 00:30:53.730 --> 00:30:58.060 and then training a model using the pGLM had

679 00:30:58.060 --> 00:31:00.100 the lowest overall holdout study error

680 00:31:00.100 --> 00:31:02.420 compared to the approach using just

681 00:31:02.420 --> 00:31:05.800 a regular generalized linear model,

682 00:31:05.800 --> 00:31:08.400 and then also the individual study approach as well.

683 00:31:09.320 --> 00:31:11.739 And we also compared it to another post called

684 00:31:11.739 --> 00:31:14.179 the Meta-Lasso, which we were able to adapt

685 00:31:14.179 --> 00:31:15.760 to do prediction, and we didn't see that much improvement

686 00:31:15.760 --> 00:31:17.000 of performance as well.

687 00:31:17.000 --> 00:31:20.640 But in general, the result that we saw here was

688 00:31:20.640 --> 00:31:23.259 that the individual study approach had

689 00:31:23.259 --> 00:31:26.570 bad prediction error also across the different studies.

690 00:31:26.570 --> 00:31:29.060 So again, this sort of takes what we've already seen

691 00:31:29.060 --> 00:31:31.190 in the literature in terms of inconsistency,

692 00:31:31.190 --> 00:31:33.330 in terms of the number of genes that are being selected

693 00:31:33.330 --> 00:31:35.140 in each of these models, and also the variations

694 00:31:35.140 --> 00:31:38.450 in the prediction accuracy, this sort of reflects

695 00:31:38.450 --> 00:31:41.523 what we've been seeing in some of this prior work.

696 00:31:43.730 --> 00:31:45.663 So in order to you implement this approach

697 00:31:45.663 --> 00:31:49.070 in a more systematic way, my student and I,

698 00:31:49.070 --> 00:31:51.427 Hillary worked, put together an R package called

699 00:31:51.427 --> 00:31:53.880 The GLMMPen R Package.

700 00:31:53.880 --> 00:31:56.050 So this was just recently submitted

701 00:31:56.050 --> 00:31:58.960 to Journal of Statistical Software, but if you want to track

702 00:31:58.960 --> 00:32:01.610 the code, it's available on Github right here,

703 00:32:01.610 --> 00:32:05.170 and we're in the process of submitting this to CRAN as well.

704 00:32:05.170 --> 00:32:07.880 This was sort of like a nice starter project that I gave

705 00:32:07.880 --> 00:32:12.030 to Hillary to, you know, get her feet wet with coding,

706 00:32:12.030 --> 00:32:14.523 and she's done a really great job, you know,

707 00:32:14.523 --> 00:32:16.280 in terms of putting this together.

708 00:32:16.280 --> 00:32:19.163 And some of the distinct differences between this

709 00:32:19.163 --> 00:32:21.360 and what we put forth in the paper is the use

710 00:32:21.360 --> 00:32:23.994 of Hamiltonian Monte Carlo and the east app,

711 00:32:23.994 --> 00:32:25.842 instead of the Metropolis Gibbs.

712 00:32:25.842 --> 00:32:26.980 It's much faster, much more efficient.

713 00:32:26.980 --> 00:32:28.674 We also have added helper functions

714 00:32:28.674 --> 00:32:32.978 for the (indistinct) tuning parameters, and also making

715 00:32:32.978 --> 00:32:35.773 some diagnostic plots as well, after convergence.

716 00:32:36.640 --> 00:32:38.670 And we've also implemented some speed

717 00:32:38.670 --> 00:32:41.470 and memory improvements as well, to help with usability.

718 00:32:44.170 --> 00:32:47.060 Okay, so we talked about some issues

719 00:32:47.060 --> 00:32:49.850 regarding data integration, and then issues

720 00:32:49.850 --> 00:32:52.490 with normalization, how that impedes, or can impede

721 00:32:52.490 --> 00:32:55.730 validation in future patients, and then we introduced

722 00:32:55.730 --> 00:32:58.680 a way to sidestep the normalization problem,

723 00:32:58.680 --> 00:33:00.890 using this sort of rank-based transformation,

724 00:33:00.890 --> 00:33:03.394 and an approach to select consistent predictors

725 00:33:03.394 --> 00:33:06.970 in the presence of between-study heterogeneity.

726 00:33:06.970 --> 00:33:09.250 So next, I'm going to talk about a case study

727 00:33:09.250 --> 00:33:12.820 in pancreatic cancer, where we took a lot of these tools,

728 00:33:12.820 --> 00:33:16.450 and applied them to a problem that some collaboratives

729 00:33:16.450 --> 00:33:20.150 of mine were having, you know, at the cancer center at UNC.

730 00:33:20.150 --> 00:33:23.370 And to give a brief overview of pancreatic cancer,

731 00:33:23.370 --> 00:33:25.850 it has a really poor prognosis.

732 00:33:25.850 --> 00:33:29.870 Five-year survival is very low, you know, typically 5%.

733 00:33:29.870 --> 00:33:32.480 The median survival tends to be less than 11 months,

734 00:33:32.480 --> 00:33:35.260 and the main reason why this is the case is that

735 00:33:35.260 --> 00:33:37.280 early detection is very difficult,

736 00:33:37.280 --> 00:33:39.890 and so when patients show up to the clinic,

737 00:33:39.890 --> 00:33:43.850 they're oftentimes in later stages, or gone metastatic.

738 00:33:43.850 --> 00:33:48.030 So for those reasons, it's really important to place

739 00:33:48.030 --> 00:33:51.040 patients on optimal therapies upfront, and choosing

740 00:33:51.040 --> 00:33:53.980 the best therapies, specifically for a patient, you know,

741 00:33:53.980 --> 00:33:55.920 when after they're diagnosed.

742 00:33:55.920 --> 00:33:58.850 So breast and colorectal cancers have

743 00:33:58.850 --> 00:34:02.350 long-established subtyping systems that are oftentimes used.

744 00:34:02.350 --> 00:34:04.130 Again, an example of a few of them in breast

745 00:34:04.130 --> 00:34:05.770 that have actually been approved by the FDA

746 00:34:05.770 --> 00:34:09.190 for clinical use, but there's nothing available for,

747 00:34:09.190 --> 00:34:11.480 in terms of precision medicine for pancreatic cancer,

748 00:34:11.480 --> 00:34:14.260 except for a couple of targeted therapies

749 00:34:14.260 --> 00:34:15.543 for specific mutations.

750 00:34:17.430 --> 00:34:19.870 So in 2015, the Yeh Lab at UNC,

751 00:34:19.870 --> 00:34:23.890 using a combination of non-negative matrix factorization

752 00:34:23.890 --> 00:34:27.480 and consensus clustering, where it was able to discover

753 00:34:27.480 --> 00:34:29.996 two potentially clinically applicable subtypes

754 00:34:29.996 --> 00:34:33.070 in pancreatic cancer, which they call basal-like,

755 00:34:33.070 --> 00:34:37.036 the orange line here, which has a much worse survival

756 00:34:37.036 --> 00:34:40.890 compared to this classical subtype in blue,

757 00:34:40.890 --> 00:34:43.677 where patients seem to do a little bit better.

758 00:34:43.677 --> 00:34:44.940 And so with this approach, they used

759 00:34:44.940 --> 00:34:48.140 this unsupervised learning, set of learning techniques

760 00:34:48.140 --> 00:34:51.010 to derive these novel subtypes.

761 00:34:51.010 --> 00:34:54.010 And so when they took these subtypes and overlaid them

762 00:34:54.010 --> 00:34:55.640 from data from a clinical trial where they had

763 00:34:55.640 --> 00:34:57.540 treatment response information, they found that

764 00:34:57.540 --> 00:35:02.280 largely patients who with basal-like subtype tended to have

765 00:35:02.280 --> 00:35:03.650 tumors that did not respond

766 00:35:03.650 --> 00:35:06.317 to common first-line therapy, Folfirinox.

767 00:35:06.317 --> 00:35:08.260 Their tumors tended to grow from baseline.

768 00:35:08.260 --> 00:35:11.920 Whereas patients that were the classical subtype tended

769 00:35:11.920 --> 00:35:15.640 to respond better on average compared to the basal samples.

770 00:35:15.640 --> 00:35:19.580 So the implications here are that if you are,

771 00:35:19.580 --> 00:35:22.680 subtype is basal, you should avoid Folfirinox

772 00:35:22.680 --> 00:35:25.020 at baseline entry with an alternative type drug,

773 00:35:25.020 --> 00:35:27.387 typically Gemcitabine and nab-paclitaxel Abraxane.

774 00:35:27.387 --> 00:35:28.740 And then for classical patients,

775 00:35:28.740 --> 00:35:30.290 they should receive Folfirinox.

776 00:35:32.114 --> 00:35:34.130 But the problem here is that subtyping clearly is

777 00:35:34.130 --> 00:35:35.540 an unsupervised learning approach, right?

778 00:35:35.540 --> 00:35:36.750 It's not a prediction tool.

779 00:35:36.750 --> 00:35:41.750 So it's, this approach is quite limited if it,

780 00:35:42.240 --> 00:35:44.970 when you have to do, assign a subtype

781 00:35:44.970 --> 00:35:47.710 in a small number of patients, it just doesn't work.

782 00:35:47.710 --> 00:35:49.610 So what some people have done in the past,

783 00:35:49.610 --> 00:35:52.220 so they simply take new patients, and recluster them

784 00:35:52.220 --> 00:35:54.570 with existing, their existing training samples.

785 00:35:54.570 --> 00:35:58.140 The problem with that is that the subtype assignments

786 00:35:58.140 --> 00:36:00.100 for those original training samples might change

787 00:36:00.100 --> 00:36:01.110 when they recluster it.

788 00:36:01.110 --> 00:36:02.660 So there's not a stable, it's not really

789 00:36:02.660 --> 00:36:04.930 a stable approach to really do this.

790 00:36:04.930 --> 00:36:07.938 So the goal here was to leverage the existing training data

791 00:36:07.938 --> 00:36:11.517 that's available to the lab, which come

792 00:36:11.517 --> 00:36:14.855 from different platforms to come up with an approach,

793 00:36:14.855 --> 00:36:17.677 a classifier to predict subtype, given

794 00:36:17.677 --> 00:36:19.930 new subtypes information, genomic,

795 00:36:19.930 --> 00:36:23.394 a new patient's genomic data, to get subtype,

796 00:36:23.394 --> 00:36:24.890 a predicted subtype for that individual.

797 00:36:24.890 --> 00:36:28.410 So of course, in that scenario, we also want to make sure

798 00:36:28.410 --> 00:36:30.670 that that process is simplified, and that we make

799 00:36:30.670 --> 00:36:32.760 this prediction process as easy as possible,

800 00:36:32.760 --> 00:36:36.157 in the face of all these issues we talked about regarding

801 00:36:36.157 --> 00:36:39.780 normalization and the training data to each other,

802 00:36:39.780 --> 00:36:42.440 and also normalization of the new patient data

803 00:36:42.440 --> 00:36:43.940 to the existing training data.

804 00:36:45.260 --> 00:36:48.820 So using some of the techniques that we just talked about,

805 00:36:48.820 --> 00:36:50.760 we came up with a classifier that we call PurIST,

806 00:36:50.760 --> 00:36:53.430 which was published in the CCR last year,

807 00:36:53.430 --> 00:36:56.270 where essentially we were able to do that.

808 00:36:56.270 --> 00:36:59.170 We take in the genomic data for a previous patient,

809 00:36:59.170 --> 00:37:04.170 and able to predict subtype based off of that,

810 00:37:04.180 --> 00:37:05.800 the train model that we developed.

811 00:37:05.800 --> 00:37:08.754 And in this particular paper, we had nine data sets

812 00:37:08.754 --> 00:37:10.750 that we curated from the literature, three of which

813 00:37:10.750 --> 00:37:12.578 that we used for training,

814 00:37:12.578 --> 00:37:13.540 the rest we used for validation.

815 00:37:13.540 --> 00:37:16.400 And we did consensus clustering on all of them,

816 00:37:16.400 --> 00:37:18.110 using the gene list that was derived

817 00:37:18.110 --> 00:37:19.623 from the original publication,

818 00:37:20.978 --> 00:37:22.800 where the subtypes were discovered to get labels,

819 00:37:22.800 --> 00:37:25.180 subject labels for each one of the subjects

820 00:37:25.180 --> 00:37:26.820 in each one of these studies.

821 00:37:26.820 --> 00:37:30.370 So once we had those labels from consensus clustering,

822 00:37:30.370 --> 00:37:33.170 we then merged the data from our three largest studies,

823 00:37:33.170 --> 00:37:34.970 which are our training studies.

824 00:37:34.970 --> 00:37:37.340 We did some sample for filtering based on quality,

825 00:37:37.340 --> 00:37:40.070 and we filtered some genes based off of, you know,

826 00:37:40.070 --> 00:37:42.440 expression levels and things like that.

827 00:37:42.440 --> 00:37:45.010 And then we applied our previous training approach

828 00:37:45.010 --> 00:37:49.917 to get a small subset of top scoring pairs from the data.

829 00:37:49.917 --> 00:37:51.230 And in this case, we have eight that we selected,

830 00:37:51.230 --> 00:37:55.430 each with their own study-level coefficient.

831 00:37:55.430 --> 00:37:57.580 And then for prediction, the process is very simple,

832 00:37:57.580 --> 00:38:00.300 we just check in that patient, whether gene A is greater

833 00:38:00.300 --> 00:38:02.130 than gene D for each of these pairs,

834 00:38:02.130 --> 00:38:05.240 and that gives us their binary vector of ones and zeros.

835 00:38:05.240 --> 00:38:08.630 We multiply that by the coefficients from the train model.

836 00:38:08.630 --> 00:38:11.460 This is basically just calculating a linear predictor

837 00:38:11.460 --> 00:38:13.750 from this logistic regression model.

838 00:38:13.750 --> 00:38:14.850 And then we can convert that

839 00:38:14.850 --> 00:38:18.130 to a predicted probability of being basal.

840 00:38:18.130 --> 00:38:23.130 So using this approach, we were able to select

841 00:38:23.130 --> 00:38:25.170 16 genes pertaining to eight subtypes,

842 00:38:25.170 --> 00:38:27.210 but we can find here that the predictions

843 00:38:27.210 --> 00:38:30.760 from this model tends to coincide very strongly

844 00:38:30.760 --> 00:38:32.930 with the labels that were collected

845 00:38:32.930 --> 00:38:33.980 using consensus clusters.

846 00:38:33.980 --> 00:38:36.498 So that gives us some confidence that reproducing

847 00:38:36.498 --> 00:38:41.070 in some way, you know, this, the result that we got

848 00:38:41.070 --> 00:38:43.100 using this clustering approach.

849 00:38:43.100 --> 00:38:46.100 You can also clearly see here that as the subtype changes,

850 00:38:46.100 --> 00:38:48.620 that you see flips in the expression in each one

851 00:38:48.620 --> 00:38:51.760 of the pairs of genes that we collected

852 00:38:51.760 --> 00:38:53.680 in this particular study.

853 00:38:53.680 --> 00:38:55.010 And then when we applied this model

854 00:38:55.010 --> 00:38:58.740 to six external validation dataset, we found that it had

855 00:38:58.740 --> 00:39:01.330 a very good performance in terms of recapitulating subtype,

856 00:39:01.330 --> 00:39:03.660 where we had a relatively good sensitivity

857 00:39:03.660 --> 00:39:07.090 and specificity in each case, which we owe part

858 00:39:07.090 --> 00:39:08.185 to the fact that we don't have to worry as much

859 00:39:08.185 --> 00:39:13.185 about this sort of cross-study normalization training time

860 00:39:13.218 --> 00:39:16.570 or test time, and also the fact that we leveraged

861 00:39:17.407 --> 00:39:18.620 multiple data sets when selecting

862 00:39:20.570 --> 00:39:21.690 the predictors for this model.

863 00:39:21.690 --> 00:39:23.870 And so when we looked at the predictive values

864 00:39:23.870 --> 00:39:26.510 in these holdout studies, the predictive subtypes,

865 00:39:26.510 --> 00:39:29.660 we recapitulated the differences in survival

866 00:39:29.660 --> 00:39:31.850 that we observed in other studies as well,

867 00:39:31.850 --> 00:39:34.354 where basal-like patients do a lot worse

868 00:39:34.354 --> 00:39:36.700 compared to classical patients.

869 00:39:36.700 --> 00:39:38.690 If you want to look a little bit more at the details

870 00:39:38.690 --> 00:39:41.100 in this paper, you can check out this link here,

871 00:39:41.100 --> 00:39:43.720 and if you want to access the code that we used

872 00:39:43.720 --> 00:39:45.460 to make these predictions, that's available

873 00:39:45.460 --> 00:39:48.453 on this Github page at this link right here.

874 00:39:50.380 --> 00:39:53.310 Another thing that we were able to show is that for patients

875 00:39:53.310 --> 00:39:56.450 that had samples that are collected through different modes

876 00:39:56.450 --> 00:40:00.070 of collection, whether it was bulk, FNA, FFPE,

877 00:40:00.070 --> 00:40:03.020 we found that the predictions in these patients tend to be

878 00:40:03.020 --> 00:40:06.430 highly consistent, and this is basically deriving

879 00:40:06.430 --> 00:40:08.820 itself, again, from the simple assumption behind TSPs,

880 00:40:08.820 --> 00:40:13.060 where the relative rank within the subject of the expression

881 00:40:13.060 --> 00:40:14.990 of these genes is predicted.

882 00:40:14.990 --> 00:40:17.310 So as long as that is being preserved,

883 00:40:17.310 --> 00:40:21.440 then you should be able to have the model predict well

884 00:40:21.440 --> 00:40:23.289 in different scenarios.

885 00:40:23.289 --> 00:40:27.630 So when we also went through CLIA validation for this tool,

886 00:40:27.630 --> 00:40:31.154 we also confirmed 95% agreement between replicated runs

887 00:40:31.154 --> 00:40:36.154 in other platforms, and we also confirmed concordance

888 00:40:37.950 --> 00:40:42.770 between NanoString and RNAC, also through different modes

889 00:40:42.770 --> 00:40:43.603 of sample collection.

890 00:40:43.603 --> 00:40:46.690 So right now this is the first clinically applicable test

891 00:40:46.690 --> 00:40:50.610 for a prospect of first line treatment selection in PDAC.

892 00:40:50.610 --> 00:40:54.250 And right now we do have a study that just recently opened

893 00:40:54.250 --> 00:40:56.390 at the Medical College of Wisconsin that's using PurIST

894 00:40:56.390 --> 00:40:58.390 for prospect of treatment selection,

895 00:40:58.390 --> 00:41:01.970 and we have another one opening at University of Rochester,

896 00:41:01.970 --> 00:41:06.320 and also at UNC soon as well.

897 00:41:06.320 --> 00:41:09.510 So this is just an example about how you can take

898 00:41:09.510 --> 00:41:14.040 a problem, you know, in, from the literature,

899 00:41:14.040 --> 00:41:17.570 from your collaborators, come up with a method,

900 00:41:17.570 --> 00:41:22.150 and some theory behind it, and really be able to come up

901 00:41:22.150 --> 00:41:24.310 with a good solution that is robust,

902 00:41:24.310 --> 00:41:27.440 and that can really help your collaborative

903 00:41:27.440 --> 00:41:29.763 at your institution and elsewhere.

904 00:41:31.850 --> 00:41:33.510 Okay, so that was the case study.

905 00:41:33.510 --> 00:41:34.560 To talk about some current work

906 00:41:34.560 --> 00:41:36.150 that we're doing just briefly.

907 00:41:36.150 --> 00:41:39.350 So we wanted to think about how we can also scale up the,

908 00:41:39.350 --> 00:41:42.200 this particular framework that we developed for the pGLMM,

909 00:41:42.200 --> 00:41:44.190 and one idea that we're pursuing right now

910 00:41:44.190 --> 00:41:46.400 with my student Hillary, is that we're thinking

911 00:41:47.773 --> 00:41:49.751 about using, borrowing ideas from factor analysis

912 00:41:49.751 --> 00:41:52.570 to decompose, do a deep, deterministic decomposition

913 00:41:52.570 --> 00:41:56.370 of the random effects to a lower dimensional space,

914 00:41:56.370 --> 00:41:59.690 where essentially, we can essentially map

915 00:41:59.690 --> 00:42:02.780 between the lower dimensional space (indistinct) factors,

916 00:42:02.780 --> 00:42:05.220 which is r-dimensional, to this higher dimensional space,

917 00:42:05.220 --> 00:42:10.220 using some by matrix B, which is q by r,

918 00:42:11.920 --> 00:42:16.050 and essentially in doing so, this reduces the dimension

919 00:42:16.050 --> 00:42:19.243 of the integral in the Monte Carlo EM algorithm.

920 00:42:20.253 --> 00:42:21.730 So rather than having to do approximate integral

921 00:42:21.730 --> 00:42:23.560 and q dimensions, which can be difficult,

922 00:42:23.560 --> 00:42:26.870 you can work in a much lower space in terms of integral,

923 00:42:26.870 --> 00:42:28.710 and then have this additional problem

924 00:42:28.710 --> 00:42:30.590 of trying to estimate this matrix,

925 00:42:30.590 --> 00:42:33.170 and not back to the original dimension cube.

926 00:42:33.170 --> 00:42:34.840 So that's something that we're just starting to work on

927 00:42:34.840 --> 00:42:38.550 right now, and another thing that we're starting to work on

928 00:42:38.550 --> 00:42:41.229 is the idea of trying to extend some of the work

929 00:42:41.229 --> 00:42:42.860 in variational autoencoders

930 00:42:42.860 --> 00:42:45.200 that my student David is working on now.

931 00:42:45.200 --> 00:42:48.253 His current work is trying to account for missing data

932 00:42:48.253 --> 00:42:51.350 when trying to train these sort of deep learning models,

933 00:42:51.350 --> 00:42:55.170 the VAEs unsupervised learning model's oftentimes used

934 00:42:55.170 --> 00:42:56.010 for dimensional reduction.

935 00:42:56.010 --> 00:42:57.020 You might've heard of it

936 00:42:57.020 --> 00:43:01.330 in single cells sequencing applications.

937 00:43:01.330 --> 00:43:02.850 But the question that we wanted to address is, well,

938 00:43:02.850 --> 00:43:04.990 what if you have missing data, you know,

939 00:43:04.990 --> 00:43:08.197 in your input features X, which might be (indistinct)?

940 00:43:09.529 --> 00:43:14.260 So essentially we were able to develop input.

941 00:43:14.260 --> 00:43:17.280 So we have a pre-print up right now, it's the code,

942 00:43:17.280 --> 00:43:20.240 and we're looking to extend this, where essentially,

943 00:43:20.240 --> 00:43:22.680 rather than worrying about this latent space Z,

944 00:43:22.680 --> 00:43:24.640 which we're assuming that that encodes a lot

945 00:43:24.640 --> 00:43:26.910 of the information in the original data,

946 00:43:26.910 --> 00:43:28.910 we replaced that with learning the posterior

947 00:43:28.910 --> 00:43:31.550 of the random effect, given the observed data.

948 00:43:31.550 --> 00:43:34.260 And then in the second portion here, we replaced

949 00:43:34.260 --> 00:43:38.820 this generative model with the general model of y given X

950 00:43:38.820 --> 00:43:40.680 in the random effects.

951 00:43:40.680 --> 00:43:42.880 So that's another avenue that can allow us

952 00:43:42.880 --> 00:43:44.650 to hopefully account for non-linearity,

953 00:43:44.650 --> 00:43:47.100 and arbitrator action between features as well.

954 00:43:47.100 --> 00:43:49.179 And also it might be an easier way to scale up

955 00:43:49.179 --> 00:43:52.570 some of the analysis we've done too,

956 00:43:52.570 --> 00:43:55.330 which I've already mentioned.

957 00:43:55.330 --> 00:43:58.361 Okay, so in terms of some concluding thoughts,

958 00:43:58.361 --> 00:44:02.762 I talked a lot about how the original subtypes were derived

959 00:44:02.762 --> 00:44:05.930 for this pancreatic cancer case study using NMF

960 00:44:05.930 --> 00:44:09.310 and consensus clustering to get two subtypes.

961 00:44:09.310 --> 00:44:12.310 But there were also other groups that are published,

962 00:44:12.310 --> 00:44:15.540 subtyping systems, that in one, they found

963 00:44:15.540 --> 00:44:19.150 three subtypes, and in another one they found four subtypes.

964 00:44:19.150 --> 00:44:22.042 So the question is, well, you know, well,

965 00:44:22.042 --> 00:44:23.270 which one do we use?

966 00:44:23.270 --> 00:44:26.130 Again, this is also confusing for practitioners

967 00:44:26.130 --> 00:44:28.950 about which approach might be more meaningful

968 00:44:28.950 --> 00:44:30.110 in the clinical setting.

969 00:44:30.110 --> 00:44:31.840 And each of these approaches were also derived

970 00:44:31.840 --> 00:44:35.480 using NMF and consensus clustering, and they were done

971 00:44:35.480 --> 00:44:37.540 separately on different patient cohorts

972 00:44:37.540 --> 00:44:39.140 at different institutions.

973 00:44:39.140 --> 00:44:41.460 So you can see that this is another reflection

974 00:44:41.460 --> 00:44:44.930 of heterogeneity in single-study learning,

975 00:44:44.930 --> 00:44:48.680 and how we can get these different or discrepant results

976 00:44:48.680 --> 00:44:52.170 from applying the same technique to 200 genus datasets

977 00:44:52.170 --> 00:44:54.400 that were generated at different places.

978 00:44:54.400 --> 00:44:57.000 So of course this creates another problem, you know,

979 00:44:57.000 --> 00:44:59.730 who's right, which approach do we use?

980 00:44:59.730 --> 00:45:03.350 And it's kind of like a circular argument here.

981 00:45:03.350 --> 00:45:06.870 So in the paper that I mentioned before with PurIST,

982 00:45:06.870 --> 00:45:09.260 another thing that we did is we overlaid

983 00:45:09.260 --> 00:45:11.839 the others subtype system calls

984 00:45:11.839 --> 00:45:14.790 with the observed clinical outcomes

985 00:45:14.790 --> 00:45:16.650 for the studies that we collected.

986 00:45:16.650 --> 00:45:19.120 And one of the things that we found was that,

987 00:45:19.120 --> 00:45:21.920 and these other subtyping systems,

988 00:45:21.920 --> 00:45:23.840 each of them also had something,

989 00:45:23.840 --> 00:45:26.990 something that was very similar to the basal-like subtype,

990 00:45:26.990 --> 00:45:29.860 and for the remaining subtypes, they had survival

991 00:45:29.860 --> 00:45:32.650 that was similar to the classical subtype.

992 00:45:32.650 --> 00:45:35.210 So one of the arguments that we made was that,

993 00:45:35.210 --> 00:45:36.813 well, if the clinical outcomes are the same

994 00:45:36.813 --> 00:45:39.570 for the other subtypes, you know,

995 00:45:39.570 --> 00:45:41.500 are they exactly right necessary

996 00:45:41.500 --> 00:45:43.250 for clinical decision-making?

997 00:45:43.250 --> 00:45:45.540 That was one argument that we put forth.

998 00:45:45.540 --> 00:45:48.420 And when we looked at the response data, again,

999 00:45:48.420 --> 00:45:51.410 we saw that one of the subtypes in the other approaches

1000 00:45:51.410 --> 00:45:56.020 also overlapped the basal-like subtype in terms of response.

1001 00:45:56.020 --> 00:45:57.430 And then for the remaining subtypes,

1002 00:45:57.430 --> 00:46:00.900 they were just kind of randomly dispersed at the other end,

1003 00:46:00.900 --> 00:46:05.280 you know, of the spectrum here in terms of tumor present,

1004 00:46:05.280 --> 00:46:06.730 tumor change after treatment.

1005 00:46:06.730 --> 00:46:09.310 So the takeaway here is that heterogeneity

1006 00:46:09.310 --> 00:46:13.660 between studies also impacts tasks in unsupervised learning,

1007 00:46:13.660 --> 00:46:16.330 like the NMF+ consensus clustering approach

1008 00:46:16.330 --> 00:46:18.000 to discover subtypes.

1009 00:46:18.000 --> 00:46:20.770 And what this also does is, as you can imagine,

1010 00:46:20.770 --> 00:46:23.690 this injects a lot of confusion into the literature,

1011 00:46:23.690 --> 00:46:27.119 and can also slow down the process of translating

1012 00:46:27.119 --> 00:46:29.980 some of these approaches to the clinic.

1013 00:46:29.980 --> 00:46:31.960 So this also underlies the need

1014 00:46:31.960 --> 00:46:35.280 for replicable cross-study sub discovery approaches,

1015 00:46:35.280 --> 00:46:40.280 for replicable approaches for unsupervised learning.

1016 00:46:40.580 --> 00:46:42.980 That's something that, you know, something that we might,

1017 00:46:42.980 --> 00:46:45.630 we hope to be working on in the future,

1018 00:46:45.630 --> 00:46:47.623 and we hope to see more work on as well.

1019 00:46:48.660 --> 00:46:52.640 So to summarize the, one of the major points

1020 00:46:52.640 --> 00:46:55.470 of this talk was to introduce and discuss, you know,

1021 00:46:55.470 --> 00:46:58.100 replicability issues in genomic prediction models,

1022 00:46:58.100 --> 00:47:01.080 supervised learning, that stems from technical,

1023 00:47:01.080 --> 00:47:03.420 and also non-technical sources.

1024 00:47:03.420 --> 00:47:06.770 We also introduced a new approach to facilitate

1025 00:47:06.770 --> 00:47:08.840 data integration and multistory learning

1026 00:47:08.840 --> 00:47:12.426 in a way that captures between-study heterogeneity,

1027 00:47:12.426 --> 00:47:15.400 and showed how this can be used for the prediction

1028 00:47:15.400 --> 00:47:20.360 of subtype for pancreatic cancer, and also introduced

1029 00:47:20.360 --> 00:47:22.522 some scalable methods and future direction

1030 00:47:22.522 --> 00:47:24.933 in replicable subtype discovery.

1031 00:47:26.350 --> 00:47:28.180 So that's it for me.

1032 00:47:28.180 --> 00:47:30.140 I just want to thank some of my faculty crowd,

1033 00:47:30.140 --> 00:47:33.050 collaboratives, Quefeng Li, Junier Oliva

1034 00:47:33.050 --> 00:47:36.750 from UNC computer science, Jen Jen Yeah

1035 00:47:36.750 --> 00:47:40.010 from surgical oncology at Lineberger, 1036 00:47:40.010 --> 00:47:42.550 Joe Ibrahim as well, UNC biostatistics,

1037 00:47:42.550 --> 00:47:45.100 and also my students, Hilary, who's done a lot of work

1038 00:47:45.100 --> 00:47:47.821 in this area, and also David Lim, who's doing

1039 00:47:47.821 --> 00:47:49.840 some of the deep learning work in our group.

1040 00:47:49.840 --> 00:47:51.283 And that's it, thank you.

1041 00:47:57.800 --> 00:47:59.290 <v Robert>So does anybody here have</v>

1042 00:47:59.290 --> 00:48:01.830 any questions for the professor?

1043 00:48:09.063 --> 00:48:14.063 Or anybody on the, on Zoom, any questions you want to ask?

1044 00:48:25.900 --> 00:48:27.383 <v ->It looks like I'm off the hook.</v>

1045 00:48:28.750 --> 00:48:30.240 <v Robert>All right, well, thank you so much.</v>

1046 00:48:30.240 --> 00:48:31.813 Really appreciated your talk.

1047 00:48:33.390 --> 00:48:34.490 Have a good afternoon.

1048 00:48:36.030 --> 00:48:37.880 <v ->All right, thank you for having me.</v>