WEBVTT

1 00:00:00.000 --> 00:00:04.167 (interference drowns out speaker)

2 00:00:08.192 --> 00:00:11.279 <v Announcer>Biostatistics, computational.</v>

3 00:00:11.279 --> 00:00:14.670 (interference drowns out speaker)

4 00:00:14.670 --> 00:00:19.486 So before joining UCLA in 2013.

5 00:00:19.486 --> 00:00:23.640 (interference drowns out speaker)

6 00:00:23.640 --> 00:00:24.660 Production work.

7 00:00:24.660 --> 00:00:28.800 (interference drowns out speaker)

8 00:00:28.800 --> 00:00:33.242 On the script is (interference drowns out speaker)

9 00:00:33.242 --> 00:00:35.317 include differentiation factors,

10 00:00:35.317 --> 00:00:37.316 asymmetric (indistinct) replication,

11 00:00:37.316 --> 00:00:39.520 p-value-free false discovery (indistinct),

12 00:00:39.520 --> 00:00:42.540 and a high dimensional variable selection.

13 00:00:42.540 --> 00:00:46.200 And on the bio (indistinct) application side,

14 00:00:46.200 --> 00:00:50.415 her research include all single cell (indistinct)

15 00:00:50.415 --> 00:00:52.031 for (indistinct) genomics and (indistinct).

16 00:00:52.031 --> 00:00:56.198 (interference drowns out speaker)

17 00:00:59.610 --> 00:01:01.331 Research published.

18 00:01:01.331 --> 00:01:05.498 (interference drowns out speaker)

19 00:01:13.971 --> 00:01:14.804 2019

20 00:01:16.170 --> 00:01:20.905 She's an MIT Technology Review certified (indistinct)

21 00:01:20.905 --> 00:01:24.679 in 2020, and she has received from Harvard.

22 00:01:24.679 --> 00:01:28.846 (interference drowns out speaker)

23 00:01:30.943 --> 00:01:33.360 <v Jingyi>I couldn't wait for the introduction.</v>

24 00:01:33.360 --> 00:01:36.240 It's my honor here to present my work,

25 00:01:36.240 --> 00:01:39.810 and my sabbatical in this fellowship program

26 00:01:39.810 --> 00:01:41.370 at Harvard Radcliffe Institute.

27 00:01:41.370 --> 00:01:46.050 So it’s my pleasure to talk about some of our recent work

28 00:01:46.050 --> 00:01:50.910 related to how statistic rigor is important in genomics.

29 00:01:50.910 --> 00:01:54.390 So I want to say that when I was a student,

30 00:01:54.390 --> 00:01:56.940 especially I think most of our audience here are students,

31 00:01:56.940 --> 00:01:59.550 I want to give you this motivation.

32 00:01:59.550 --> 00:02:02.040 When I was a student back in 2007,

33 00:02:02.040 --> 00:02:05.340 that was when I just started my PhD

34 00:02:05.340 --> 00:02:08.010 and I was interested in bioinformatics.

35 00:02:08.010 --> 00:02:11.280 I had a lot of questions about bioinformatics methods

36 00:02:11.280 --> 00:02:13.740 after I took statistics classes.

37 00:02:13.740 --> 00:02:16.020 I think some of the questions I listed here

38 00:02:16.020 --> 00:02:18.540 include are P values valid?

39 00:02:18.540 --> 00:02:20.910 Because P values are so widely used

40 00:02:20.910 --> 00:02:22.800 in genomics bioinformatics.

41 00:02:22.800 --> 00:02:24.930 And also, we have a lot

42 00:02:24.930 --> 00:02:28.380 of bio bioinformatics methods developed for data analysis.

43 00:02:28.380 --> 00:02:31.680 And I wonder why don’t we use classical statistical methods

44 00:02:31.680 --> 00:02:32.550 in textbooks?

45 00:02:32.550 --> 00:02:33.840 And the third thing is,

46 00:02:33.840 --> 00:02:38.280 when we use statistical test to understand the question,

47 00:02:38.280 --> 00:02:39.750 to answer some pivot question,

48 00:02:39.750 --> 00:02:41.880 what is the proper null hypothesis?

49 00:02:41.880 --> 00:02:44.760 So you will see those questions in the topics

50 00:02:44.760 --> 00:02:46.710 I will talk about next.

51 00:02:46.710 --> 00:02:51.510 So this talk will focus on the multiple testing problem.

52 00:02:51.510 --> 00:02:53.220 See, multiple testing, what it means

53 00:02:53.220 --> 00:02:56.730 is that we have multiple hypothesis tests,

54 00:02:56.730 --> 00:03:00.660 and the criteria we use in this problem are P values,

55 00:03:00.660 --> 00:03:04.170 which we have one P value per test.

56 00:03:04.170 --> 00:03:07.620 So we know that the requirement for a valid P value

57 00:03:07.620 --> 00:03:11.490 is that P values should follow the uniform distribution

58 00:03:11.490 --> 00:03:14.460 between zero one under the null hypothesis.

59 00:03:14.460 --> 00:03:17.640 Or we may relax this to be super uniform.

60 00:03:17.640 --> 00:03:18.900 Just for your information,

61 00:03:18.900 --> 00:03:21.780 super uniform means that the P values

62 00:03:21.780 --> 00:03:24.900 have higher density toward one

63 00:03:24.900 --> 00:03:26.430 and lower density towards zero.

64 00:03:26.430 --> 00:03:29.760 So that's still okay for type one error control,

65 00:03:29.760 --> 00:03:31.290 even though you may have a larger

66 00:03:31.290 --> 00:03:33.000 than expected type two error.

67 00:03:33.000 --> 00:03:35.010 So given the many, many P values,

68 00:03:35.010 --> 00:03:39.900 we need one criterion to set a cutoff on the P values.

69 00:03:39.900 --> 00:03:41.970 And the most commonly used criterion

70 00:03:41.970 --> 00:03:43.800 for multiple testing correction

71 00:03:43.800 --> 00:03:46.950 is called a false discovery rate, short as FPR.

72 00:03:46.950 --> 00:03:51.600 So the definition here is the expectation of this ratio,

73 00:03:51.600 --> 00:03:55.410 and this ratio is the number of false discoveries

74 00:03:55.410 --> 00:03:57.270 over the number of discoveries.

75 00:03:57.270 --> 00:04:00.090 So this notation means the maximum

76 00:04:00.090 --> 00:04:02.280 between the number of discoveries and one.

77 00:04:02.280 --> 00:04:05.640 In other words, we don't allow the denominator to be zero,

78 00:04:05.640 --> 00:04:07.200 if we don't make any discovery.

79 00:04:07.200 --> 00:04:09.750 So this is to avoid the dividing zero issue.

80 00:04:09.750 --> 00:04:13.650 And this ratio has a name called false discovery proportion.

81 00:04:13.650 --> 00:04:15.720 In other words, we can have this proportion

82 00:04:15.720 --> 00:04:18.030 for one particular data set.

83 00:04:18.030 --> 00:04:21.660 However, as you know, we don't observe this ratio

84 00:04:21.660 --> 00:04:24.510 because we don't know which discoveries are false.

85 00:04:24.510 --> 00:04:27.780 So therefore, this ratio is only a hypothetical concept,

86 00:04:27.780 --> 00:04:30.420 but not really computable.

87 00:04:30.420 --> 00:04:31.920 And here, the expectation

88 00:04:31.920 --> 00:04:35.100 is taken over all possible data set

89 00:04:35.100 --> 00:04:38.130 from the same distribution as our data set.

90 00:04:38.130 --> 00:04:40.260 So this is the frequentist concept

91 00:04:40.260 --> 00:04:43.980 because we have imaginary potential data sets.

92 00:04:43.980 --> 00:04:46.950 So therefore, the phenomena paper

93 00:04:46.950 --> 00:04:49.830 by Benjamini and Hochburg gave us a way

94 00:04:49.830 --> 00:04:53.280 to control this expectation called FDR

95 00:04:53.280 --> 00:04:56.730 under a claimed level, say, 5%,

96 00:04:56.730 --> 00:05:00.600 even though we couldn't realize this ratio itself.

97 00:05:00.600 --> 00:05:02.400 But we could control its expectation.

98 00:05:02.400 --> 00:05:04.830 So that's the magic of statistics.

99 00:05:04.830 --> 00:05:07.020 So Benjamini Hochburg algorithm allows us

100 00:05:07.020 --> 00:05:11.190 to set a cutoff on the P values to control the FDR.

101 00:05:11.190 --> 00:05:14.790 But I want to emphasize that the FDS's only controlled

102 00:05:14.790 --> 00:05:17.280 when P values satisfy this assumption,

103 00:05:17.280 --> 00:05:19.320 otherwise, it may not be.

104 00:05:19.320 --> 00:05:24.320 So I want to say three common causes of ill-posed P values,

105 00:05:24.360 --> 00:05:27.480 which make P values don't satisfy this assumption

106 00:05:27.480 --> 00:05:30.217 in genomics, and I'll go through them one by one.

107 00:05:31.110 --> 00:05:34.170 The first issue is what I call the formulation

108 00:05:34.170 --> 00:05:37.740 of a two sample test problem as a one sample test.

109 00:05:37.740 --> 00:05:39.060 What does this mean?

110 00:05:39.060 --> 00:05:42.090 So I will use the common genomic analysis

111 00:05:42.090 --> 00:05:44.670 of ChIP-seq data as an example.

112 00:05:44.670 --> 00:05:45.990 So in ChIP-seq data,

113 00:05:45.990 --> 00:05:50.160 we want to measure where a protein binds in the genome.

114 00:05:50.160 --> 00:05:53.520 So you can consider the X axis as the genome

115 00:05:53.520 --> 00:05:56.790 and the Y axis as the protein binding intensity

116 00:05:56.790 --> 00:05:58.680 measured by ChIP-seq.

117 00:05:58.680 --> 00:06:02.070 So here, we have experimental sample,

118 00:06:02.070 --> 00:06:05.550 the condition of our interest, say, a certain cell line.

119 00:06:05.550 --> 00:06:08.040 And the background sample is what we know

120 00:06:08.040 --> 00:06:09.660 that there's no protein,

121 00:06:09.660 --> 00:06:11.790 so there should be no protein binding.

122 00:06:11.790 --> 00:06:15.420 But we still want to measure the noise from the experiment.

123 00:06:15.420 --> 00:06:17.430 So we need this contrast.

124 00:06:17.430 --> 00:06:21.900 And here, we want to say that the region in the red box,

125 00:06:21.900 --> 00:06:25.500 this interval, we want to call it as a peak,

126 00:06:25.500 --> 00:06:29.550 if we see the intensity in the experimental sample

127 00:06:29.550 --> 00:06:32.790 is much larger than the intensity in the background sample.

128 00:06:32.790 --> 00:06:35.940 So we do the comparison and we want to cut this at a peak.

129 00:06:35.940 --> 00:06:38.820 That's the purpose of this analysis.

130 00:06:38.820 --> 00:06:41.550 And I wanna say that, in the field,

131 00:06:41.550 --> 00:06:45.390 because ChIP-seq has become popular since 2008,

132 00:06:45.390 --> 00:06:49.290 Macs and Homer are probably the two most popular software

133 00:06:49.290 --> 00:06:50.940 for cutting peaks.

134 00:06:50.940 --> 00:06:53.850 Even though they have very complex procedures

135 00:06:53.850 --> 00:06:56.460 for processing the sequencing data

136 00:06:56.460 --> 00:06:58.380 that in a statistical part

137 00:06:58.380 --> 00:07:00.960 to call a region as a peak or not,

138 00:07:00.960 --> 00:07:04.140 I can say, their formulation is as follows.

139 00:07:04.140 --> 00:07:08.580 Given a region, we count its number of ChIP-seq reads

140 00:07:08.580 --> 00:07:12.210 in the background sample and in the experimental sample.

141 00:07:12.210 --> 00:07:15.450 So let's just summarize this intensity as a count,

142 00:07:15.450 --> 00:07:19.140 a count here, a count here, and both are now negative.

143 00:07:19.140 --> 00:07:21.270 So I call the background count as big X,

144 00:07:21.270 --> 00:07:23.580 experimental count as big Y.

145 00:07:23.580 --> 00:07:27.120 And in our data, we have the observations, right?

146 00:07:27.120 --> 00:07:30.180 We refer to them as small x, small y.

147 00:07:30.180 --> 00:07:33.330 Then, the P value in both software

148 00:07:33.330 --> 00:07:36.840 is essentially this probability, the probability

149 00:07:36.840 --> 00:07:41.840 that big Y is greater or equal than the observed small y,

150 00:07:41.910 --> 00:07:45.240 where the big Y follows upon some distribution

151 00:07:45.240 --> 00:07:48.240 with mean parameter as the small x.

152 00:07:48.240 --> 00:07:51.090 Now, when I look at this formula back in 2008,

153 00:07:51.090 --> 00:07:54.633 the Macs paper, I wonder whether this is correct.

154 00:07:55.620 --> 00:07:57.090 And I don't think so.

155 00:07:57.090 --> 00:07:58.950 Because the reason, if you look at it,

156 00:07:58.950 --> 00:08:00.900 is what is the null hypothesis?

157 00:08:00.900 --> 00:08:03.990 The null hypothesis is essentially, okay,

158 00:08:03.990 --> 00:08:05.700 let's assume the experimental count

159 00:08:05.700 --> 00:08:08.610 is our test statistic, okay?

160 00:08:08.610 --> 00:08:11.310 We assume it follows a Poisson distribution

161 00:08:11.310 --> 00:08:12.960 with mean lambda.

162 00:08:12.960 --> 00:08:17.960 And here, the null hypothesis is lambda is equal to small x.

163 00:08:18.090 --> 00:08:21.150 Alternative is lambda greater than small x.

164 00:08:21.150 --> 00:08:23.160 So what's the problem with here?

165 00:08:23.160 --> 00:08:27.240 Essentially, we are using small x as a fixed parameter

166 00:08:27.240 --> 00:08:29.280 instead of a random observation.

167 00:08:29.280 --> 00:08:30.270 So in other words,

168 00:08:30.270 --> 00:08:33.390 the randomness in the background count is ignored.

169 00:08:33.390 --> 00:08:36.720 We only consider experimental count as the random variable.

170 00:08:36.720 --> 00:08:39.990 So in other words, where use the two sample testing problem

171 00:08:39.990 --> 00:08:41.970 to a one sample testing problem

172 00:08:41.970 --> 00:08:43.637 because we only consider the randomness

173 00:08:43.637 --> 00:08:45.960 in the experimental sample.

174 00:08:45.960 --> 00:08:50.040 But this is not something our textbook teaches us.

175 00:08:50.040 --> 00:08:52.830 The reason is because if we consider background

176 00:08:52.830 --> 00:08:56.130 as one condition, experimental has another condition,

177 00:08:56.130 --> 00:08:59.640 under each condition, our sample size is only one.

178 00:08:59.640 --> 00:09:02.070 So therefore, the T test will not apply

179 00:09:02.070 --> 00:09:04.950 because a central limit here clearly doesn't apply.

180 00:09:04.950 --> 00:09:08.733 So how do we calculate P value, any ideas?

181 00:09:09.600 --> 00:09:12.750 I think one possibility that we could still assume

182 00:09:12.750 --> 00:09:15.660 Poisson distribution for both background X

183 00:09:15.660 --> 00:09:16.830 and experimental Y.

184 00:09:16.830 --> 00:09:20.520 You have two Poisson, under the independence,

185 00:09:20.520 --> 00:09:22.560 we can probably derive the distribution

186 00:09:22.560 --> 00:09:25.620 for Y minus X, right, and what's the null distribution.

187 00:09:25.620 --> 00:09:26.760 That's the only way.

188 00:09:26.760 --> 00:09:30.000 But, if you think about it, how can we verify

189 00:09:30.000 --> 00:09:32.583 whether the Poisson distribution is reasonable?

190 00:09:32.583 --> 00:09:34.890 You only have one observation from it.

191 00:09:34.890 --> 00:09:37.140 The distribution could be anything, right?

192 00:09:37.140 --> 00:09:40.950 So assuming a parametric distribution seems quite,

193 00:09:40.950 --> 00:09:42.300 I will say, aggressive.

194 00:09:42.300 --> 00:09:45.090 So I think P value calculation is challenging here.

195 00:09:45.090 --> 00:09:48.600 And also, I even wonder, in this case,

196 00:09:48.600 --> 00:09:51.000 for this one versus one comparison,

197 00:09:51.000 --> 00:09:53.100 should we use a P value?

198 00:09:53.100 --> 00:09:57.060 Or is this really a testing problem that's feasible?

199 00:09:57.060 --> 00:09:59.940 So I would say, over the years, I gradually realized

200 00:09:59.940 --> 00:10:02.670 that here we looked at many, many regions,

201 00:10:02.670 --> 00:10:04.230 not just one region.

202 00:10:04.230 --> 00:10:08.160 So the goal or the criterion that's ultimately used

203 00:10:08.160 --> 00:10:09.240 is actually FDR.

204 00:10:09.240 --> 00:10:12.180 And in this process,

205 00:10:12.180 --> 00:10:15.960 P values are just intermediate for FDR control,

206 00:10:15.960 --> 00:10:18.180 instead of our final target.

207 00:10:18.180 --> 00:10:21.090 So do we have to stick with P values?

208 00:10:21.090 --> 00:10:25.110 This motivated me to write this paper with my students

209 00:10:25.110 --> 00:10:30.110 to propose a way to achieve p-value-free FDR control

210 00:10:30.180 --> 00:10:34.230 by leveraging the theory in Barber and Candes paper,

211 00:10:34.230 --> 00:10:35.610 their knockoff paper,

212 00:10:35.610 --> 00:10:38.580 so we could actually doing FDR control

213 00:10:38.580 --> 00:10:41.190 in this example without using P value.

214 00:10:41.190 --> 00:10:43.170 So I will talk about this later in my talk,

215 00:10:43.170 --> 00:10:46.830 but this is one motivation for the Clipper paper.

216 00:10:46.830 --> 00:10:49.950 The second issue with P values is that we observe,

217 00:10:49.950 --> 00:10:51.680 sometimes, P values are not valid

218 00:10:51.680 --> 00:10:56.680 because the parametric model used may not fit the data well.

219 00:10:57.000 --> 00:11:00.600 So this is an example for this commonly used

220 00:11:00.600 --> 00:11:04.620 differential expression analysis on RNA sequencing data.

221 00:11:04.620 --> 00:11:06.600 So for this task,

222 00:11:06.600 --> 00:11:09.750 the two popular softwares are DESeq2 and edgeR.

223 00:11:09.750 --> 00:11:12.420 So the data usually looks like this.

224 00:11:12.420 --> 00:11:15.390 So we want to compare two conditions

225 00:11:15.390 --> 00:11:18.780 and seeing which genes are differentially expressed.

226 00:11:18.780 --> 00:11:21.630 So condition one, we have three samples,

227 00:11:21.630 --> 00:11:23.400 which we cause to replicate,

228 00:11:23.400 --> 00:11:25.410 condition two, three replicates.

229 00:11:25.410 --> 00:11:29.070 So every row is one replicate,

230 00:11:29.070 --> 00:11:31.170 while every column is one gene.

231 00:11:31.170 --> 00:11:33.900 So to call a gene as differentially expressed,

232 00:11:33.900 --> 00:11:36.000 we need to compare its three values

233 00:11:36.000 --> 00:11:39.150 from condition one, two, three values from condition two.

234 00:11:39.150 --> 00:11:42.960 So clearly, we can see the left one may be a D gene,

235 00:11:42.960 --> 00:11:45.060 the right one may not be a D gene, right?

236 00:11:45.060 --> 00:11:46.170 That's our intuition.

237 00:11:46.170 --> 00:11:49.470 And we want to make this more formal

238 00:11:49.470 --> 00:11:51.854 by doing a statistical test.

239 00:11:51.854 --> 00:11:54.660 But in both edgeR and DESeq2,

240 00:11:54.660 --> 00:11:58.590 you can see that to compensate the small sample size,

241 00:11:58.590 --> 00:12:00.120 like three versus three,

242 00:12:00.120 --> 00:12:05.120 they assume a gene follows a negative binomial distribution

243 00:12:05.460 --> 00:12:06.630 under each condition.

244 00:12:06.630 --> 00:12:09.300 So essentially, these three values are assumed

245 00:12:09.300 --> 00:12:12.120 to follow one negative binomial distribution.

246 00:12:12.120 --> 00:12:13.380 These three values

247 00:12:13.380 --> 00:12:16.170 follow another negative binomial distribution.

248 00:12:16.170 --> 00:12:17.640 And the null hypothesis

249 00:12:17.640 --> 00:12:20.700 is the two negative binomial distributions

250 00:12:20.700 --> 00:12:23.550 have the same mean, that's the problem.

251 00:12:23.550 --> 00:12:27.090 Okay, so we actually discovered an issue

252 00:12:27.090 --> 00:12:29.850 with popular methods from this data set.

253 00:12:29.850 --> 00:12:32.306 And thanks to my collaborator Dr. Wei Li

254 00:12:32.306 --> 00:12:35.520 who is a computation of biologist at UC Irvine.

255 00:12:35.520 --> 00:12:39.480 So actually, from this patient data,

256 00:12:39.480 --> 00:12:43.110 we have a much larger sample size, 51 patients

257 00:12:43.110 --> 00:12:46.645 before the treatment of some immunotherapy medicine,

258 00:12:46.645 --> 00:12:49.620 58 patients on treatment.

259 00:12:49.620 --> 00:12:52.680 So we want to compare the RNA sequencing data

260 00:12:52.680 --> 00:12:54.930 of these two groups of patients.

261 00:12:54.930 --> 00:12:59.930 So essentially, when we apply DESeq2 or edgeR to this data,

262 00:13:00.840 --> 00:13:05.253 the red dots indicate the number of D genes identified.

263 00:13:06.300 --> 00:13:11.300 To verify whether we can still identify D genes

264 00:13:11.640 --> 00:13:12.840 from permuted data,

265 00:13:12.840 --> 00:13:15.150 because the reason is that we want to see

266 00:13:15.150 --> 00:13:18.780 whether the permuted data is actually really,

267 00:13:18.780 --> 00:13:20.220 because we know the permuted data

268 00:13:20.220 --> 00:13:21.840 shouldn't give us any signals.

269 00:13:21.840 --> 00:13:23.820 If we just disrupt the two groups,

270 00:13:23.820 --> 00:13:25.830 we shouldn't expect any D genes.

271 00:13:25.830 --> 00:13:29.070 But surprisingly, we found that each method

272 00:13:29.070 --> 00:13:33.540 can identify sometimes even more D genes from permuted data.

273 00:13:33.540 --> 00:13:37.230 So the bar and the error bars show the distribution

274 00:13:37.230 --> 00:13:40.350 of D genes identified from permuted data.

275 00:13:40.350 --> 00:13:43.710 So this is something quite unexpected.

276 00:13:43.710 --> 00:13:46.530 And to look into the reason, our first thought

277 00:13:46.530 --> 00:13:49.920 is to check the negative binomial assumption.

278 00:13:49.920 --> 00:13:51.780 Because now, under each group,

279 00:13:51.780 --> 00:13:54.900 we have 51 and 58 sample sizes,

280 00:13:54.900 --> 00:13:58.680 so we could check the distribution, and here's what we get.

281 00:13:58.680 --> 00:14:02.753 You see that for the genes that are frequently identified

282 00:14:02.753 --> 00:14:06.990 from permuted data, if we run the goodness-of-fit test,

283 00:14:06.990 --> 00:14:09.870 we check the negative binomial distribution,

284 00:14:09.870 --> 00:14:12.360 these genes have very small P values,

285 00:14:12.360 --> 00:14:15.090 indicating that this fit is not good.

286 00:14:15.090 --> 00:14:16.350 Well, if you look at the genes

287 00:14:16.350 --> 00:14:19.950 that are rarely identified from permuted data,

288 00:14:19.950 --> 00:14:22.860 the P values are bigger and the goodness-of-fit is better.

289 00:14:22.860 --> 00:14:25.200 So we do see this relationship

290 00:14:25.200 --> 00:14:28.740 between the goodness-of-fit of negative binomial

291 00:14:28.740 --> 00:14:31.590 and the frequency that a gene is identified

292 00:14:31.590 --> 00:14:33.240 from permuted data.

293 00:14:33.240 --> 00:14:36.480 So negative binomial model seems to not fit well

294 00:14:36.480 --> 00:14:39.030 on this patient data.

295 00:14:39.030 --> 00:14:42.090 Because here, the 51 patients shouldn't be regarded

296 00:14:42.090 --> 00:14:44.700 as replicates, they're not experimental replicates,

297 00:14:44.700 --> 00:14:46.110 they are individuals.

298 00:14:46.110 --> 00:14:49.590 So therefore, the theory for deriving negative binomials

299 00:14:49.590 --> 00:14:52.440 usually assume as a Gamma-Poisson Mixture model,

300 00:14:52.440 --> 00:14:54.180 Gamma-Poisson Hierarchical model.

301 00:14:54.180 --> 00:14:56.580 That one may no longer hold,

302 00:14:56.580 --> 00:14:59.610 and that's why we think the parametric model

303 00:14:59.610 --> 00:15:03.510 is not applicable to this patient data.

304 00:15:03.510 --> 00:15:05.580 So what's the consequence, right?

305 00:15:05.580 --> 00:15:07.650 So we want to convince the scientist

306 00:15:07.650 --> 00:15:10.530 what's the consequence of doing this analysis

307 00:15:10.530 --> 00:15:12.090 in this problematic way.

308 00:15:12.090 --> 00:15:14.910 We show that if we just use the D genes

309 00:15:14.910 --> 00:15:17.070 found by DESeq2 and edgeR,

310 00:15:17.070 --> 00:15:20.233 which are the genes corresponding to the red dot,

311 00:15:20.233 --> 00:15:23.460 around the so called gene oncology analysis,

312 00:15:23.460 --> 00:15:26.130 that is to check which functional terms

313 00:15:26.130 --> 00:15:29.370 are enriched in those two gene sets,

314 00:15:29.370 --> 00:15:31.350 we can see many functional terms

315 00:15:31.350 --> 00:15:33.510 are related to immune functions.

316 00:15:33.510 --> 00:15:35.730 Which would suggest that if we trust

317 00:15:35.730 --> 00:15:38.820 these two methods' results, we may conclude that,

318 00:15:38.820 --> 00:15:41.400 yes, between the two groups of patients,

319 00:15:41.400 --> 00:15:44.430 there are differences in immune responses, right?

320 00:15:44.430 --> 00:15:48.030 That seems to confirm our scientific hypothesis.

321 00:15:48.030 --> 00:15:50.610 However, now, we see many of these genes

322 00:15:50.610 --> 00:15:53.790 were also identified from permuted data,

323 00:15:53.790 --> 00:15:57.120 then, that will make the results dubious.

324 00:15:57.120 --> 00:16:01.470 So what we tried is that, even the sample size is so large,

325 00:16:01.470 --> 00:16:03.690 we tried the classical Wilcoxon rank sign test,

326 00:16:03.690 --> 00:16:05.240 which everybody learned, right?

327 00:16:06.119 --> 00:16:08.310 So non parametric two sample test

328 00:16:08.310 --> 00:16:11.130 that doesn't assume a parametric distribution.

329 00:16:11.130 --> 00:16:13.080 And here, it's self consistent,

330 00:16:13.080 --> 00:16:16.590 it doesn't identify D genes from real data,

331 00:16:16.590 --> 00:16:20.040 but also, it doesn't identify D genes from permuted data.

332 00:16:20.040 --> 00:16:22.650 So there's no contradiction here.

333 00:16:22.650 --> 00:16:25.860 And this result motivated me to ask this question,

334 00:16:25.860 --> 00:16:27.603 which I had years ago,

335 00:16:28.590 --> 00:16:32.730 should we always use popular bioinformatics tools?

336 00:16:32.730 --> 00:16:35.010 Like, check the citation of these two methods,

337 00:16:35.010 --> 00:16:36.213 super highly cited.

338 00:16:37.080 --> 00:16:39.150 Should I reuse popular method

339 00:16:39.150 --> 00:16:42.990 or should we consider general statistical methods,

340 00:16:42.990 --> 00:16:44.760 like Wilcoxon.

341 00:16:44.760 --> 00:16:49.650 So our recommendation is sample size matters, right?

342 00:16:49.650 --> 00:16:52.380 We may have different methods

343 00:16:52.380 --> 00:16:54.660 suitable for different sample sizes,

344 00:16:54.660 --> 00:16:57.510 and essentially, why statistics has so many methods,

345 00:16:57.510 --> 00:16:58.923 paramedic, non parametric,

346 00:16:59.910 --> 00:17:02.880 is because we have different scenarios in our data.

347 00:17:02.880 --> 00:17:04.740 That's the first thing we should realize.

348 00:17:04.740 --> 00:17:07.500 It's not like one method can do all the things.

349 00:17:07.500 --> 00:17:10.140 And the second thing is sanity check.

350 00:17:10.140 --> 00:17:12.770 We should always consider doing some sanity check

351 00:17:12.770 --> 00:17:14.760 to make sure we trust the results

352 00:17:14.760 --> 00:17:17.460 instead of just take the results for granted.

353 00:17:17.460 --> 00:17:20.370 So these things were summarized in our paper

354 00:17:20.370 --> 00:17:22.920 published earlier this year.

355 00:17:22.920 --> 00:17:24.660 And since its publication,

356 00:17:24.660 --> 00:17:27.960 we have received a lot of discussions on Twitter,

357 00:17:27.960 --> 00:17:29.010 if you are interested.

358 00:17:29.010 --> 00:17:31.800 But anyway, so it means that many people are interested

359 00:17:31.800 --> 00:17:35.940 in this topic, especially many people, users believe

360 00:17:35.940 --> 00:17:39.377 that popular bioinformatics tools are the state-of-the-art,

361 00:17:39.377 --> 00:17:41.985 right, the way, standard methods (indistinct).

362 00:17:41.985 --> 00:17:45.420 But if you are bio statisticians, you may not like this.

363 00:17:45.420 --> 00:17:47.760 Because we want to develop new methods.

364 00:17:47.760 --> 00:17:49.500 Otherwise, what's our job, right?

365 00:17:49.500 --> 00:17:53.400 So in this case, we need to really find the loopholes,

366 00:17:53.400 --> 00:17:57.090 or the limitations, or the gap between current approach

367 00:17:57.090 --> 00:17:58.410 and the data scenarios,

368 00:17:58.410 --> 00:18:00.900 and try convinces people that, yes,

369 00:18:00.900 --> 00:18:03.570 we do need careful thoughts when we choose method.

370 00:18:03.570 --> 00:18:06.240 It's not always one method.

371 00:18:06.240 --> 00:18:08.280 And a related question is,

372 00:18:08.280 --> 00:18:12.720 in Wilcoxon, definitely doesn't have a strong assumption,

373 00:18:12.720 --> 00:18:15.120 and (indistinct) have a reasonable power

374 00:18:15.120 --> 00:18:16.920 when the sample size is large.

375 00:18:16.920 --> 00:18:19.770 But what if sample sizes are small, right?

376 00:18:19.770 --> 00:18:21.450 So when it's small, we know,

377 00:18:21.450 --> 00:18:24.750 non parametric tests like Wilcoxon doesn't have power.

378 00:18:24.750 --> 00:18:29.670 So in this case, we actually proposed Clipper again,

379 00:18:29.670 --> 00:18:34.050 so it can work as a downstream correction tool

380 00:18:34.050 --> 00:18:36.300 for DESeq2 and edgeR.

381 00:18:36.300 --> 00:18:38.700 Because they are supposed to be quite powerful,

382 00:18:38.700 --> 00:18:41.010 even though they find probably too many.

383 00:18:41.010 --> 00:18:44.190 So hopefully, we could use that to borrow their power,

384 00:18:44.190 --> 00:18:47.310 but help them improve the FDR control.

385 00:18:47.310 --> 00:18:50.310 So I'll show the results later in my talk.

386 00:18:50.310 --> 00:18:51.630 That's the second cause.

387 00:18:51.630 --> 00:18:53.760 And the third cause for ill-posed P values

388 00:18:53.760 --> 00:18:55.950 is a little more complicated.

389 00:18:55.950 --> 00:18:59.670 And this is the issue commonly observed in single cell data,

390 00:18:59.670 --> 00:19:01.080 single cell RNA-seq data.

391 00:19:01.080 --> 00:19:02.910 So I will use this analysis

392 00:19:02.910 --> 00:19:07.910 called pseudotime differentially expressed genes as example.

393 00:19:08.190 --> 00:19:09.858 What is a pseudotime?

394 00:19:09.858 --> 00:19:13.110 Pseudotime means it's not real time, it's pseudo, right?

395 00:19:13.110 --> 00:19:15.720 So it's something we inferred

396 00:19:15.720 --> 00:19:17.670 from single cell RNA-seq data,

397 00:19:17.670 --> 00:19:20.430 so those cells are measured all at once.

398 00:19:20.430 --> 00:19:25.430 But we want to infer some time trajectory from the cells.

399 00:19:25.920 --> 00:19:28.830 So I'll just use the screenshot from Slingshot,

400 00:19:28.830 --> 00:19:33.830 which is a method for inferring pseudotime for explanation.

401 00:19:34.050 --> 00:19:39.050 So here, this is a two-dimensional PCA plot of cells,

402 00:19:39.180 --> 00:19:41.280 and the cells are pre-clustered,

403 00:19:41.280 --> 00:19:44.250 so each color represents one cluster.

404 00:19:44.250 --> 00:19:47.100 So the Slingshot algorithm does the following,

405 00:19:47.100 --> 00:19:50.610 first, it takes the cluster means' centers,

406 00:19:50.610 --> 00:19:52.770 and connect them using the algorithm

407 00:19:52.770 --> 00:19:54.450 called minimum spanning tree.

408 00:19:54.450 --> 00:19:55.730 So if you're not familiar with that,

409 00:19:55.730 --> 00:19:59.370 it has an equivalence with hierarchical clustering actually.

410 00:19:59.370 --> 00:20:02.400 So with the minimum spanning tree, you get this tree,

411 00:20:02.400 --> 00:20:06.840 and then, they smooth out the tree using principle curves.

412 00:20:06.840 --> 00:20:08.130 So we have two curves,

413 00:20:08.130 --> 00:20:09.810 and then for every cell,

414 00:20:09.810 --> 00:20:13.080 we find the closest curve and project the cell to the curve.

415 00:20:13.080 --> 00:20:14.970 So therefore, in each curve,

416 00:20:14.970 --> 00:20:18.090 the projections are called pseudotime values.

417 00:20:18.090 --> 00:20:21.060 And usually, it's normalized between zero and one,

418 00:20:21.060 --> 00:20:23.580 so we need to find the root and call it zero,

419 00:20:23.580 --> 00:20:25.410 the other end is called one.

420 00:20:25.410 --> 00:20:28.260 So this whole process is called pseudotime inference.

421 00:20:28.260 --> 00:20:31.680 In other words, after it, we will give every cell

422 00:20:31.680 --> 00:20:35.250 a pseudotime value in each trajectory.

423 00:20:35.250 --> 00:20:37.650 Okay, so one thing I want to emphasize

424 00:20:37.650 --> 00:20:40.200 is that in this pseudotime inference

425 00:20:40.200 --> 00:20:43.470 we used gene expression values already.

426 00:20:43.470 --> 00:20:46.860 So it's not like we observe pseudotime as external variable,

427 00:20:46.860 --> 00:20:48.930 but it's from the same data.

428 00:20:48.930 --> 00:20:53.130 So I want to show what we could do after the pseudotime.

429 00:20:53.130 --> 00:20:55.560 So a typical analysis is to identify

430 00:20:55.560 --> 00:20:57.870 which genes are differentially expressed

431 00:20:57.870 --> 00:20:59.250 along the pseudotime.

432 00:20:59.250 --> 00:21:03.360 Like the left one, we see, it has this upward trajectory,

433 00:21:03.360 --> 00:21:05.910 so we may call it differentially expressed.

434 00:21:05.910 --> 00:21:08.970 And here, we want to say the pseudotime

435 00:21:08.970 --> 00:21:11.910 represent some cell immune response,

436 00:21:11.910 --> 00:21:13.560 and this is an immuno-related gene,

437 00:21:13.560 --> 00:21:16.740 so we expect to see the upward trajectory.

438 00:21:16.740 --> 00:21:20.340 For the right gene, we expect to see something constant,

439 00:21:20.340 --> 00:21:23.340 so we don't want to come right (indistinct) a D gene,

440 00:21:23.340 --> 00:21:25.350 that's the intuition.

441 00:21:25.350 --> 00:21:28.320 And I want to say that we must realize,

442 00:21:28.320 --> 00:21:31.050 pseudotime values are random

443 00:21:31.050 --> 00:21:34.770 simply because the cells is a random sample, right?

444 00:21:34.770 --> 00:21:36.600 We need to consider randomness,

445 00:21:36.600 --> 00:21:40.770 and we want to show this to people by doing subsampling.

446 00:21:40.770 --> 00:21:43.290 So you can see that sampling variation

447 00:21:43.290 --> 00:21:45.810 would get into pseudotime values.

448 00:21:45.810 --> 00:21:47.880 Here, every row is a cell.

449 00:21:47.880 --> 00:21:49.680 If I randomly subsample,

450 00:21:49.680 --> 00:21:53.310 say, 80% of cells from the left cells

451 00:21:53.310 --> 00:21:56.760 and redo the pseudotime trajectory inference,

452 00:21:56.760 --> 00:22:00.600 we can see that for the cells in the subsamples

453 00:22:00.600 --> 00:22:04.380 that include it, its values will vary to some degree.

454 00:22:04.380 --> 00:22:06.630 So it's not a constant.

455 00:22:06.630 --> 00:22:09.690 Okay, so realizing this, we should consider

456 00:22:09.690 --> 00:22:12.810 the randomness of pseudotime from the data.

457 00:22:12.810 --> 00:22:15.930 However, existing methods all treat pseudotime

458 00:22:15.930 --> 00:22:17.790 as an observed covariate.

459 00:22:17.790 --> 00:22:21.690 So our goal here is to fix this,

460 00:22:21.690 --> 00:22:24.870 and we proposed this method called PseudotimeDE,

461 00:22:24.870 --> 00:22:27.240 which actually does the inference,

462 00:22:27.240 --> 00:22:29.460 which infers whether one gene

463 00:22:29.460 --> 00:22:32.310 is differentially expressed along pseudotime,

464 00:22:32.310 --> 00:22:36.450 and by considering pseudotime inference uncertainty.

465 00:22:36.450 --> 00:22:40.620 So what we did exactly is that, here,

466 00:22:40.620 --> 00:22:43.830 to see whether a gene changes with pseudotime,

467 00:22:43.830 --> 00:22:45.480 what's the intuition?

468 00:22:45.480 --> 00:22:48.270 We should do regression, right, do a regression analysis

469 00:22:48.270 --> 00:22:52.530 by treating a gene's expression value as Y,

470 00:22:52.530 --> 00:22:54.660 pseudotime as X, and regular regression.

471 00:22:54.660 --> 00:22:57.540 Yeah, this is exactly what existing methods did.

472 00:22:57.540 --> 00:22:59.430 And to make sure the regression

473 00:22:59.430 --> 00:23:01.740 is not restricted to be linear,

474 00:23:01.740 --> 00:23:04.920 and also account for that the gene expression values

475 00:23:04.920 --> 00:23:06.570 are non negative counts.

476 00:23:06.570 --> 00:23:11.570 So actually, we choose the generalized additive model,

477 00:23:11.610 --> 00:23:14.100 which is also used in an existing method,

478 00:23:14.100 --> 00:23:15.750 which I will show very soon.

479 00:23:15.750 --> 00:23:20.400 So this is a very flexible and interpretable model.

480 00:23:20.400 --> 00:23:24.060 So generalized means Y can be non Gaussian

481 00:23:24.060 --> 00:23:25.230 and the other distribution,

482 00:23:25.230 --> 00:23:27.510 just like generalized linear model.

483 00:23:27.510 --> 00:23:28.740 But additive means

484 00:23:28.740 --> 00:23:32.340 that we make the linear model more general,

485 00:23:32.340 --> 00:23:36.553 so every feature can be non linearly transformed,

486 00:23:37.860 --> 00:23:41.550 but the features after transformations are still added.

487 00:23:41.550 --> 00:23:44.340 So that's additive, short as GAM.

488 00:23:44.340 --> 00:23:47.310 So essentially, once we have a set of cells,

489 00:23:47.310 --> 00:23:49.350 we first infer the pseudotime,

490 00:23:49.350 --> 00:23:52.170 so we order the cells along the pseudotime,

491 00:23:52.170 --> 00:23:53.430 and for gene J,

492 00:23:53.430 --> 00:23:56.550 we check how the gene changes with pseudotime,

493 00:23:56.550 --> 00:23:59.760 so we run the generalized additive model

494 00:23:59.760 --> 00:24:01.500 to obtain a test statistic.

495 00:24:01.500 --> 00:24:05.190 Please know that generalized additive model has its theory,

496 00:24:05.190 --> 00:24:08.250 so we could use the theory to calculate

497 00:24:08.250 --> 00:24:11.817 to use the null distribution and calculate P value.

498 00:24:11.817 --> 00:24:14.670 And that was done in an existing method.

499 00:24:14.670 --> 00:24:17.520 We want say that this may be problematic

500 00:24:17.520 --> 00:24:19.784 because this whole null distribution

501 00:24:19.784 --> 00:24:22.440 considers pseudotime to be fixed.

502 00:24:22.440 --> 00:24:23.820 So to address this,

503 00:24:23.820 --> 00:24:26.370 we need to consider pseudotime inference

504 00:24:26.370 --> 00:24:30.060 as part of our test statistic calculation.

505 00:24:30.060 --> 00:24:33.090 So to do this, we use the top part.

506 00:24:33.090 --> 00:24:35.023 We actually do subsapling of the cells.

507 00:24:36.270 --> 00:24:38.430 The reason we didn't do bootstrap

508 00:24:38.430 --> 00:24:41.280 is simply because we want the method to be flexible

509 00:24:41.280 --> 00:24:43.290 for pseudotime inference method.

510 00:24:43.290 --> 00:24:47.070 Like I show here, there are Slingshot, Monocle3,

511 00:24:47.070 --> 00:24:48.300 and a few others.

512 00:24:48.300 --> 00:24:49.800 We want it to be flexible,

513 00:24:49.800 --> 00:24:53.520 and some methods don't allow cells to be repetitive,

514 00:24:53.520 --> 00:24:55.710 so bootstrap doesn't apply here.

515 00:24:55.710 --> 00:24:59.370 And we use subsampling with percentage pretty high,

516 00:24:59.370 --> 00:25:03.330 like 80%, 90%, and we did a robustness analysis.

517 00:25:03.330 --> 00:25:08.010 And then, on each subsample, we do pseudotime inference.

518 00:25:08.010 --> 00:25:11.130 With this, how do we get a null distribution

519 00:25:11.130 --> 00:25:12.330 of the test statistic?

520 00:25:12.330 --> 00:25:14.700 What we did is to permute the cells,

521 00:25:14.700 --> 00:25:17.730 so any relationship between the gene J

522 00:25:17.730 --> 00:25:19.830 and the pseudotime is disrupted.

523 00:25:19.830 --> 00:25:21.563 So this can be considered from the null,

524 00:25:21.563 --> 00:25:25.410 and then, we did the same GAM model,

525 00:25:25.410 --> 00:25:29.520 and then, we calculate the values of the test statistic

526 00:25:29.520 --> 00:25:31.500 on these permuted subsamples,

527 00:25:31.500 --> 00:25:33.210 that gave us a null distribution.

528 00:25:33.210 --> 00:25:36.540 So together, we can get a P value, this is what we did.

529 00:25:36.540 --> 00:25:38.490 And we can show that this approach

530 00:25:38.490 --> 00:25:41.370 indeed can control the P values,

531 00:25:41.370 --> 00:25:44.820 make the P values uniformly distributed on the null,

532 00:25:44.820 --> 00:25:47.190 while the existing method that uses GAM,

533 00:25:47.190 --> 00:25:50.130 but only the theoretical distribution called tradeSeq,

534 00:25:50.130 --> 00:25:53.160 they have some distortion for P values.

535 00:25:53.160 --> 00:25:56.126 And then, you may wonder, what's the consequence?

536 00:25:56.126 --> 00:25:58.161 We can show that, oh, and I should say,

537 00:25:58.161 --> 00:26:03.161 Monocle3 uses generalized linear model and not uncertainty.

538 00:26:03.450 --> 00:26:06.683 So you can see that even though it's not as bad as tradeSeq,

539 00:26:06.683 --> 00:26:08.670 still, some distortion.

540 00:26:08.670 --> 00:26:09.540 So we wanna show

541 00:26:09.540 --> 00:26:13.170 that by calibrating the P value using our way

542 00:26:13.170 --> 00:26:16.740 we can actually discover more functional terms

543 00:26:16.740 --> 00:26:18.510 in our differentially expressed genes.

544 00:26:18.510 --> 00:26:21.780 It means that we can find some new biological functions

545 00:26:21.780 --> 00:26:23.730 that were missed by this new method.

546 00:26:23.730 --> 00:26:28.080 Which shows that FDR control not just help with FDR control

547 00:26:28.080 --> 00:26:29.160 of P value calibration,

548 00:26:29.160 --> 00:26:31.110 not just help with FDR control,

549 00:26:31.110 --> 00:26:33.033 but may also boost some power.

550 00:26:34.230 --> 00:26:37.290 So I just quickly talk about this PseudotimeDE,

551 00:26:37.290 --> 00:26:40.200 but I want to say that its computational time

552 00:26:40.200 --> 00:26:42.150 is the biggest limitation.

553 00:26:42.150 --> 00:26:46.485 Because here, our P value calculation requires many rounds

554 00:26:46.485 --> 00:26:50.430 of subsampling, pseudotime inference, and permutation.

555 00:26:50.430 --> 00:26:54.630 So let's say we want the P value with resolution 0.001,

556 00:26:54.630 --> 00:26:58.230 we need at least 1000 rounds of such things, right?

557 00:26:58.230 --> 00:26:59.580 That will take time.

558 00:26:59.580 --> 00:27:00.900 So the natural question

559 00:27:00.900 --> 00:27:04.410 is can we reduce the number of rounds, right,

560 00:27:04.410 --> 00:27:06.330 and still achieve FDR control?

561 00:27:06.330 --> 00:27:08.127 That becomes similar to my first goal.

562 00:27:08.127 --> 00:27:10.920 Can we get rid of the higher resolution P values,

563 00:27:10.920 --> 00:27:14.460 control the FDR, and then, we will use Clipper again.

564 00:27:14.460 --> 00:27:15.360 So you can see,

565 00:27:15.360 --> 00:27:18.120 Clipper is used throughout all the motivations,

566 00:27:18.120 --> 00:27:19.740 that's why we proposed it,

567 00:27:19.740 --> 00:27:22.140 and I'll talk about it in the next minute.

568 00:27:22.140 --> 00:27:24.330 And the second question we didn't address

569 00:27:24.330 --> 00:27:28.740 is that what if the cells don't follow a trajectory at all?

570 00:27:28.740 --> 00:27:31.590 So clearly in our null hypothesis,

571 00:27:31.590 --> 00:27:34.050 we are assuming there is a trajectory,

572 00:27:34.050 --> 00:27:38.100 it's just that gene J doesn't change with the trajectory.

573 00:27:38.100 --> 00:27:40.320 But what if the trajectory doesn't exist?

574 00:27:40.320 --> 00:27:44.580 So this whole idea of this trajectory pseudotime inference

575 00:27:44.580 --> 00:27:45.810 doesn't make sense, right?

576 00:27:45.810 --> 00:27:47.190 We need to consider that.

577 00:27:47.190 --> 00:27:50.460 But I don't think we have a good way to do it,

578 00:27:50.460 --> 00:27:53.640 unless we can change the cells to have a null

579 00:27:53.640 --> 00:27:56.490 where the cells don't follow a trajectory.

580 00:27:56.490 --> 00:27:59.070 So this motivated us to generate cells

581 00:27:59.070 --> 00:28:02.250 that don't follow a trajectory, and we used a simulator.

582 00:28:02.250 --> 00:28:05.730 So which it will be the last part I will talk about today.

583 00:28:05.730 --> 00:28:08.970 Okay, PseudotimeDE is one such a problem

584 00:28:08.970 --> 00:28:11.820 where pseudotime is inferred from the same data.

585 00:28:11.820 --> 00:28:16.820 Another common problem is to do clustering on single cells

586 00:28:17.370 --> 00:28:19.110 to identify cell clusters,

587 00:28:19.110 --> 00:28:21.060 and between cell clusters,

588 00:28:21.060 --> 00:28:23.400 we identify differentially expressed genes.

589 00:28:23.400 --> 00:28:25.710 We call this problem ClusterDE.

590 00:28:25.710 --> 00:28:29.430 But this is also using the data twice, right?

591 00:28:29.430 --> 00:28:32.400 So people have called this term double dipping,

592 00:28:32.400 --> 00:28:36.330 meaning that the same data used for twice.

593 00:28:36.330 --> 00:28:37.830 To tackle this problem,

594 00:28:37.830 --> 00:28:41.301 we need to consider the uncertainty in cell clustering,

595 00:28:41.301 --> 00:28:43.590 and there are three existing papers

596 00:28:43.590 --> 00:28:45.690 that try to address this problem

597 00:28:45.690 --> 00:28:48.480 that they either need to assume a distribution,

598 00:28:48.480 --> 00:28:52.680 like genes follow Gaussian distribution in every cluster

599 00:28:52.680 --> 00:28:56.550 or every gene follows a Poisson distribution here

600 00:28:56.550 --> 00:28:58.530 and they need to do count splitting.

601 00:28:58.530 --> 00:29:01.530 So I won't talk into the couple of details here,

602 00:29:01.530 --> 00:29:02.430 but I just want to say

603 00:29:02.430 --> 00:29:05.340 that the count splitting approach in my opinion

604 00:29:05.340 --> 00:29:07.200 tackles a different problem.

605 00:29:07.200 --> 00:29:10.710 It is conditional on the observed data matrix,

606 00:29:10.710 --> 00:29:12.660 rather than considered to be random.

607 00:29:12.660 --> 00:29:14.580 But I will not talk about the detail here.

608 00:29:14.580 --> 00:29:16.600 So motivated by the challenge in this problem,

609 00:29:16.600 --> 00:29:21.600 and we want to propose something not distribution-specific.

610 00:29:22.200 --> 00:29:27.200 We want to use our simulator to generate the null data

611 00:29:27.960 --> 00:29:31.470 and then use Clipper to achieve the FDR control.

612 00:29:31.470 --> 00:29:34.050 So we want to do this non parametrically.

613 00:29:34.050 --> 00:29:36.480 So I think the idea was motivated

614 00:29:36.480 --> 00:29:39.390 by two phenomenal statistical papers.

615 00:29:39.390 --> 00:29:41.520 One is the gap statistic paper,

616 00:29:41.520 --> 00:29:45.260 which was proposed to find the number of clusters

617 00:29:45.260 --> 00:29:46.800 in the clustering problem.

618 00:29:46.800 --> 00:29:49.440 And if you read a paper, I think the smart idea there

619 00:29:49.440 --> 00:29:53.940 is they try to generate data points without clusters

620 00:29:53.940 --> 00:29:55.590 as the negative control.

621 00:29:55.590 --> 00:29:59.160 Then, you can control your number of clusters

622 00:29:59.160 --> 00:30:00.870 with some statistic,

623 00:30:00.870 --> 00:30:03.027 versus what if there's no clusters, right,

624 00:30:03.027 --> 00:30:04.920 and do the comparison and find the gap.

625 00:30:04.920 --> 00:30:06.180 That's the gap statistic.

626 00:30:06.180 --> 00:30:08.760 And knockoffs gave the theoretical foundation

627 00:30:08.760 --> 00:30:12.393 for FDR control without using high resolution P values.

628 00:30:13.230 --> 00:30:15.600 Okay, so the halftime summary

629 00:30:15.600 --> 00:30:17.970 is that I talked about three common causes

630 00:30:17.970 --> 00:30:19.470 of ill-posed P values.

631 00:30:19.470 --> 00:30:20.970 Hopefully, I have convinced you

632 00:30:20.970 --> 00:30:24.600 that we need something to avoid this problem.

633 00:30:24.600 --> 00:30:26.220 So I talked about Clipper,

634 00:30:26.220 --> 00:30:29.730 the p-value-free FDR control for genomic feature screening.

635 00:30:29.730 --> 00:30:33.030 And as I said, it was motivated and enabled

636 00:30:33.030 --> 00:30:36.240 by the FDR control procedure from this paper.

637 00:30:36.240 --> 00:30:39.120 But the difference here is that we focus

638 00:30:39.120 --> 00:30:42.030 on marginal screening of interesting features.

639 00:30:42.030 --> 00:30:45.450 So in other words, we look at one feature at a time.

640 00:30:45.450 --> 00:30:47.190 In my previous examples,

641 00:30:47.190 --> 00:30:50.760 a feature could be a region or a gene.

642 00:30:50.760 --> 00:30:53.220 So in the original knockoff paper,

643 00:30:53.220 --> 00:30:57.240 their goal is to generate knockoff data

644 00:30:57.240 --> 00:31:01.170 just like fake data for multiple features jointly.

645 00:31:01.170 --> 00:31:02.940 And that's the very challenging part.

646 00:31:02.940 --> 00:31:05.190 But in our case, we don't need that

647 00:31:05.190 --> 00:31:07.440 because we are looking at one feature at a time,

648 00:31:07.440 --> 00:31:09.600 so it's not a multi-varied problem,

649 00:31:09.600 --> 00:31:11.610 but it's a marginal screening problem.

650 00:31:11.610 --> 00:31:15.420 So our goal is to get rid of high resolution P values.

651 00:31:15.420 --> 00:31:16.910 So the advantage of this

652 00:31:16.910 --> 00:31:20.340 is we don't need parametric distribution assumptions,

653 00:31:20.340 --> 00:31:22.410 or we don't need large sample sizes

654 00:31:22.410 --> 00:31:25.890 to enable non parametric tests, these are not needed.

655 00:31:25.890 --> 00:31:29.070 We just need to summarize every feature

656 00:31:29.070 --> 00:31:31.470 into a contrast score,

657 00:31:31.470 --> 00:31:34.500 and then, set a cutoff on the contrast scores.

658 00:31:34.500 --> 00:31:36.900 So what do I mean by contrast score?

659 00:31:36.900 --> 00:31:39.630 So every feature, say, I have total d features,

660 00:31:39.630 --> 00:31:43.380 they have C, D, sorry, d contrast scores

661 00:31:43.380 --> 00:31:45.240 shown as C1 to Cd,

662 00:31:45.240 --> 00:31:47.479 so I'm calling the histogram

663 00:31:47.479 --> 00:31:49.650 of the distribution of contrast scores.

664 00:31:49.650 --> 00:31:53.790 So if the theoretical assumption is satisfied,

665 00:31:53.790 --> 00:31:56.550 then the features that are null features

666 00:31:56.550 --> 00:32:00.210 should follow a symmetrical distribution

667 00:32:00.210 --> 00:32:01.950 around the zero, okay?

668 00:32:01.950 --> 00:32:04.200 And for the features that are interesting

669 00:32:04.200 --> 00:32:05.610 and should be discovered,

670 00:32:05.610 --> 00:32:08.610 should be large and positive on the right tail.

671 00:32:08.610 --> 00:32:12.090 So the theory of the FDR control just says,

672 00:32:12.090 --> 00:32:15.930 we can find the contrast score cutoff as t,

673 00:32:15.930 --> 00:32:20.070 such that this ratio is controlled under q.

674 00:32:20.070 --> 00:32:22.620 We ought to find the minimum t for this.

675 00:32:22.620 --> 00:32:25.902 What this means is can you can consider this ratio

676 00:32:25.902 --> 00:32:29.017 as a rough estimator of FDR.

677 00:32:29.999 --> 00:32:33.177 So the denominator is just the left tail,

678 00:32:33.177 --> 00:32:35.163 the red part plus one,

679 00:32:36.060 --> 00:32:38.910 sorry, the numerator is the right tail plus one,

680 00:32:38.910 --> 00:32:43.166 the denominator is the, sorry, the left tail is, sorry,

681 00:32:43.166 --> 00:32:45.420 the numerator is the left tail plus one,

682 00:32:45.420 --> 00:32:49.290 the denominator is the right tail with maximum with one.

683 00:32:49.290 --> 00:32:52.470 So in other words, still trying to avoid dividing zero.

684 00:32:52.470 --> 00:32:56.130 And the idea is that we want to find a threshold t,

685 00:32:56.130 --> 00:32:59.190 so that the right tail will be called discoveries

686 00:32:59.190 --> 00:33:03.330 and the left tail represent false discoveries.

687 00:33:03.330 --> 00:33:04.680 That's the intuition.

688 00:33:04.680 --> 00:33:07.770 Because we know, if the feature's null,

689 00:33:07.770 --> 00:33:11.340 then it will be randomly positive or negative.

690 00:33:11.340 --> 00:33:14.700 And the sign is independent of the absolute value.

691 00:33:14.700 --> 00:33:18.330 So that just replaces

692 00:33:18.330 --> 00:33:21.840 the uniform distribution requirement for P values,

693 00:33:21.840 --> 00:33:23.580 we change that to symmetry.

694 00:33:23.580 --> 00:33:26.100 And another thing is that the feature,

695 00:33:26.100 --> 00:33:29.490 if it's large positive, we want to discover it, right?

696 00:33:29.490 --> 00:33:31.440 So this will be the discovery set

697 00:33:31.440 --> 00:33:36.060 and this represents the negative, false discovery set.

698 00:33:36.060 --> 00:33:40.620 So that's the idea intuition behind this approach.

699 00:33:40.620 --> 00:33:42.480 But the theory to really prove it,

700 00:33:42.480 --> 00:33:45.630 we need to use Martingale in probability to prove it.

701 00:33:45.630 --> 00:33:46.980 And some of the technique was used

702 00:33:46.980 --> 00:33:48.990 for the Benjamini Hochburg procedure

703 00:33:48.990 --> 00:33:50.460 still based on Martingale.

704 00:33:50.460 --> 00:33:54.360 So anyway, this allows us to really control the FDR

705 00:33:54.360 --> 00:33:55.860 just using contrast scores.

706 00:33:55.860 --> 00:33:58.110 And another thing I found as appealing

707 00:33:58.110 --> 00:34:01.020 is that if you visually inspect the contract scores,

708 00:34:01.020 --> 00:34:05.100 you can see whether the assumption seems to be reasonable

709 00:34:05.100 --> 00:34:07.680 because you expect to see something symmetrical

710 00:34:07.680 --> 00:34:09.810 plus a heavy right tail.

711 00:34:09.810 --> 00:34:13.800 Okay, so we are currently writing to make this more formal,

712 00:34:13.800 --> 00:34:15.000 so we could actually check

713 00:34:15.000 --> 00:34:18.150 whether the assumption is reasonably holding.

714 00:34:18.150 --> 00:34:19.470 So with this approach,

715 00:34:19.470 --> 00:34:24.470 we can make a lot of the comparison analysis easier

716 00:34:25.560 --> 00:34:29.550 because the key is to find a reasonable contrast score

717 00:34:29.550 --> 00:34:31.830 that satisfies this assumption.

718 00:34:31.830 --> 00:34:35.070 And I can say that there may be multiple contrast scores

719 00:34:35.070 --> 00:34:37.410 that satisfy, not just the unique one.

720 00:34:37.410 --> 00:34:39.550 Then the difference is power, right?

721 00:34:39.550 --> 00:34:41.160 So we may have a better power

722 00:34:41.160 --> 00:34:44.250 if you have a heavier right tail.

723 00:34:44.250 --> 00:34:47.040 Okay, so for a ChIP-seq peak calling analysis,

724 00:34:47.040 --> 00:34:49.230 we can say that the contrast score

725 00:34:49.230 --> 00:34:51.870 will be comparing the target data

726 00:34:51.870 --> 00:34:54.630 from experimental condition to the null data,

727 00:34:54.630 --> 00:34:56.370 which is the background condition.

728 00:34:56.370 --> 00:34:59.280 They serve a natural pair of contrast,

729 00:34:59.280 --> 00:35:03.390 and we could apply any pipeline to each data,

730 00:35:03.390 --> 00:35:05.633 the same pipeline and then do the contrast, right?

731 00:35:05.633 --> 00:35:08.130 You can imagine, if there's no peak,

732 00:35:08.130 --> 00:35:09.690 then these two values will be,

733 00:35:09.690 --> 00:35:12.690 which one is bigger is equally likely.

734 00:35:12.690 --> 00:35:15.720 And for the RNA-seq analysis,

735 00:35:15.720 --> 00:35:20.280 here, I showed we could use permuted data as the null data

736 00:35:20.280 --> 00:35:21.840 actual data as a target data.

737 00:35:21.840 --> 00:35:25.020 So if we run some test on actual data

738 00:35:25.020 --> 00:35:26.550 to get a test statistic,

739 00:35:26.550 --> 00:35:28.560 we use the same test on permuted data

740 00:35:28.560 --> 00:35:32.190 to get a test statistic, and they serve as a contrast.

741 00:35:32.190 --> 00:35:34.890 And finally, for the PseudotimeDE and ClusterDE,

742 00:35:34.890 --> 00:35:36.660 the single cell problem,

743 00:35:36.660 --> 00:35:40.050 actual data will give us some comparison,

744 00:35:40.050 --> 00:35:41.670 either PseudotimeDE

745 00:35:41.670 --> 00:35:45.750 or the between ClusterDE test statistic.

746 00:35:45.750 --> 00:35:48.150 And if we have some similar data

747 00:35:48.150 --> 00:35:49.440 that represents the null,

748 00:35:49.440 --> 00:35:51.900 like null trajectory, null cluster,

749 00:35:51.900 --> 00:35:55.290 we could run the same pipeline and then do the contrast.

750 00:35:55.290 --> 00:35:57.180 So you see, this actually free us

751 00:35:57.180 --> 00:36:00.237 from saying we need to derive P values

752 00:36:00.237 --> 00:36:02.160 and we need to know the distribution

753 00:36:02.160 --> 00:36:05.340 by either theory or by numerical simulation, right?

754 00:36:05.340 --> 00:36:06.450 These are all relieved

755 00:36:06.450 --> 00:36:08.250 because we just need to do a contrast.

756 00:36:08.250 --> 00:36:11.670 And the power is gained from the many, many tests,

757 00:36:11.670 --> 00:36:12.930 we look at them together.

758 00:36:12.930 --> 00:36:13.763 So that's why

759 00:36:13.763 --> 00:36:15.137 this idea (background noise drowns out speaker).

760 00:36:16.080 --> 00:36:19.950 Okay, so as I said, we tried to implement Clipper

761 00:36:19.950 --> 00:36:22.380 as a way to improve FDR control,

762 00:36:22.380 --> 00:36:23.880 and we did achieve this

763 00:36:23.880 --> 00:36:27.240 for the popular software Macs and Homer

764 00:36:27.240 --> 00:36:29.070 for ChIP-seq peak calling

765 00:36:29.070 --> 00:36:32.820 and DESeq2 to edgeR for RNA-seq DEG identification.

766 00:36:32.820 --> 00:36:36.660 So you see that they did have inflated FDR,

767 00:36:36.660 --> 00:36:39.300 so the Y axis is the actual FDR,

768 00:36:39.300 --> 00:36:41.940 X axis is the target FDR threshold.

769 00:36:41.940 --> 00:36:43.733 There are inflations,

770 00:36:43.733 --> 00:36:46.410 but with our Clipper as an add-on

771 00:36:46.410 --> 00:36:48.750 to be used downstream of what they output

772 00:36:48.750 --> 00:36:50.430 and do the contrast,

773 00:36:50.430 --> 00:36:53.610 we can largely reduce the FDR to the target

774 00:36:53.610 --> 00:36:56.340 and still maintain quite good power.

775 00:36:56.340 --> 00:36:59.073 So that's the usage of Clipper as and add-on.

776 00:36:59.910 --> 00:37:01.890 And for the single cell part,

777 00:37:01.890 --> 00:37:04.710 I didn't finish about the null data generation.

778 00:37:04.710 --> 00:37:06.000 How do we do it?

779 00:37:06.000 --> 00:37:09.510 Our simulator was proposed partly for this reason,

780 00:37:09.510 --> 00:37:11.520 but it has more uses.

781 00:37:11.520 --> 00:37:14.730 So I just want to say that it's called scDesign3

782 00:37:14.730 --> 00:37:18.570 because we have scDesign and scDesign2 as two previous work.

783 00:37:18.570 --> 00:37:19.830 Now, focus on scDesign2

784 00:37:19.830 --> 00:37:23.580 because it is the direct predecessor of scDesign3.

785 00:37:23.580 --> 00:37:25.650 So what scDesign2 two does

786 00:37:25.650 --> 00:37:30.480 is it tries to fit a multi-gene probabilistic model

787 00:37:30.480 --> 00:37:32.370 for each cell type,

788 00:37:32.370 --> 00:37:35.430 and then, every gene assumes to follow

789 00:37:35.430 --> 00:37:39.000 a parametric distribution within the cell type.

790 00:37:39.000 --> 00:37:40.950 And the major contribution

791 00:37:40.950 --> 00:37:43.650 is that we capture gene-gene correlations

792 00:37:43.650 --> 00:37:45.150 using Gaussian copula.

793 00:37:45.150 --> 00:37:47.430 That will make the data more realistic.

794 00:37:47.430 --> 00:37:48.990 Here is the comparison.

795 00:37:48.990 --> 00:37:51.780 This is the real data used for fitting the model.

796 00:37:51.780 --> 00:37:54.960 This is the lab (indistinct) test data used for validation,

797 00:37:54.960 --> 00:37:58.860 and this is the synthetic cells using copula.

798 00:37:58.860 --> 00:38:01.710 If we remove the copula, the cells will look like this.

799 00:38:01.710 --> 00:38:03.600 So not realistic at all.

800 00:38:03.600 --> 00:38:07.770 And our data is more realistic than other simulators

801 00:38:07.770 --> 00:38:12.030 that did not explicitly capture gene-gene correlation.

802 00:38:12.030 --> 00:38:14.340 Although, they have some implicit mechanism,

803 00:38:14.340 --> 00:38:16.710 but the model is different.

804 00:38:16.710 --> 00:38:21.300 Okay, so we realize that scDesign2 is doing a good job

805 00:38:21.300 --> 00:38:22.830 for displaying cell types,

806 00:38:22.830 --> 00:38:24.750 but it cannot generate data like this

807 00:38:24.750 --> 00:38:26.940 from a continuous trajectory.

808 00:38:26.940 --> 00:38:30.960 What we could do is to force the cells to be divided

809 00:38:30.960 --> 00:38:32.400 and then use scDesign2.

810 00:38:32.400 --> 00:38:35.490 But then, you can see the cells are kind of in clusters,

811 00:38:35.490 --> 00:38:36.930 right, not in real data.

812 00:38:36.930 --> 00:38:40.620 But with our generalization to the version three,

813 00:38:40.620 --> 00:38:45.090 we now can generate cells from a continuous trajectory.

814 00:38:45.090 --> 00:38:48.480 And I can quickly say that we basically generalize this,

815 00:38:48.480 --> 00:38:51.180 this count distribution per cell type

816 00:38:51.180 --> 00:38:54.600 to a generalized additive model, which I already said.

817 00:38:54.600 --> 00:38:57.270 So we could make it more flexible in general,

818 00:38:57.270 --> 00:39:01.530 and scDesign2 becomes a special case of scDesign3.

819 00:39:01.530 --> 00:39:03.020 And one more thing we could do

820 00:39:03.020 --> 00:39:06.270 is we actually use the technique vine copula,

821 00:39:06.270 --> 00:39:11.070 so we could get the likelihood of how the model fits

822 00:39:11.070 --> 00:39:15.180 to the real data, so we can get the likelihood of the model,

823 00:39:15.180 --> 00:39:18.060 which can also give us more information.

824 00:39:18.060 --> 00:39:21.420 So besides the single cell trajectory data,

825 00:39:21.420 --> 00:39:24.720 we can also use the idea to generate spatial data.

826 00:39:24.720 --> 00:39:27.660 So here, the modification is that for every gene

827 00:39:27.660 --> 00:39:31.680 we assume a Gaussian process in the 2D space,

828 00:39:31.680 --> 00:39:33.630 so it can have a smooth function

829 00:39:33.630 --> 00:39:35.580 for (indistinct) expression (indistinct).

830 00:39:35.580 --> 00:39:40.020 And also, my other student help with making the simulator

831 00:39:40.020 --> 00:39:44.220 to generate reads, sequencing reads, not just counts.

832 00:39:44.220 --> 00:39:46.080 So we can go from counts to reads,

833 00:39:46.080 --> 00:39:48.450 and this will give us more functionality

834 00:39:48.450 --> 00:39:51.240 to benchmark some low level tools.

835 00:39:51.240 --> 00:39:52.380 So in short,

836 00:39:52.380 --> 00:39:55.920 the scDesign3 simulator has two functionalities.

837 00:39:55.920 --> 00:39:58.590 One is to do, of course, simulation.

838 00:39:58.590 --> 00:40:02.070 We can generate single cell data from cell types,

839 00:40:02.070 --> 00:40:04.740 discrete, continuous trajectories,

840 00:40:04.740 --> 00:40:06.990 or even in the spatial domain.

841 00:40:06.990 --> 00:40:09.172 We could generate feature modalities

842 00:40:09.172 --> 00:40:11.617 we call multi-omics, including RNA-seq,

843 00:40:11.617 --> 00:40:13.920 ATAC-seq, which is a technology

844 00:40:13.920 --> 00:40:16.020 for open chromatin measurement,

845 00:40:16.020 --> 00:40:19.350 CITE-seq, which includes both protein and RNA,

846 00:40:19.350 --> 00:40:21.030 and also DNA methylation.

847 00:40:21.030 --> 00:40:24.120 These are the examples we tried, but we could do even more.

848 00:40:24.120 --> 00:40:27.960 We could allow it to generate data with experimental designs

849 00:40:27.960 --> 00:40:32.960 including sample covariate, conditions, or even batches.

850 00:40:33.120 --> 00:40:36.150 So these can make us generate cases

851 00:40:36.150 --> 00:40:38.760 for more types of benchmarking.

852 00:40:38.760 --> 00:40:41.160 And for interpreting real data,

853 00:40:41.160 --> 00:40:44.730 scDesign3 can give us model parameters,

854 00:40:44.730 --> 00:40:47.400 so we can know whether a gene has different means

855 00:40:47.400 --> 00:40:48.990 in two cell types,

856 00:40:48.990 --> 00:40:51.900 whether a gene has a certain change on a pseudotime,

857 00:40:51.900 --> 00:40:54.930 or a gene has a certain change in two dimensional space.

858 00:40:54.930 --> 00:40:56.100 And also, as I said,

859 00:40:56.100 --> 00:40:58.980 we can output a likelihood that can give us a way

860 00:40:58.980 --> 00:41:02.580 to calculate the basic information criterion BIC,

861 00:41:02.580 --> 00:41:03.960 so we could evaluate

862 00:41:03.960 --> 00:41:07.230 whether some pseudotime describes data well,

863 00:41:07.230 --> 00:41:09.050 whether the algorithm for pseudotime inference

864 00:41:09.050 --> 00:41:10.800 does a good job,

865 00:41:10.800 --> 00:41:13.260 or whether the clusters explain data well.

866 00:41:13.260 --> 00:41:14.850 So these are the things we could do.

867 00:41:14.850 --> 00:41:17.910 And finally, to generate the null data

868 00:41:17.910 --> 00:41:19.770 for the Clipper (indistinct),

869 00:41:19.770 --> 00:41:22.200 we can alter the model parameters.

870 00:41:22.200 --> 00:41:25.080 Like this is what we fit from real data,

871 00:41:25.080 --> 00:41:27.300 we could change the model parameters

872 00:41:27.300 --> 00:41:30.360 to make the gene no longer differentially expressed,

873 00:41:30.360 --> 00:41:32.580 have the same mean in two subtypes.

874 00:41:32.580 --> 00:41:34.650 Or, after we fit a real data

875 00:41:34.650 --> 00:41:36.900 with two cell types or two clusters,

876 00:41:36.900 --> 00:41:39.870 we could change the cluster parameter

877 00:41:39.870 --> 00:41:42.450 to make sure the cells come from one cluster

878 00:41:42.450 --> 00:41:43.920 instead of two clusters.

879 00:41:43.920 --> 00:41:46.680 So these are the things we could do with the model.

880 00:41:46.680 --> 00:41:49.140 And so this is how our paper,

881 00:41:49.140 --> 00:41:52.230 but more details are in our paper, which has been posted,

882 00:41:52.230 --> 00:41:53.910 if you are interested.

883 00:41:53.910 --> 00:41:55.710 And I want to just quickly show

884 00:41:55.710 --> 00:41:58.830 how the ClusterDE analysis could be done.

885 00:41:58.830 --> 00:42:01.350 This is the real data with two clusters.

886 00:42:01.350 --> 00:42:03.030 I want to say that this is the case

887 00:42:03.030 --> 00:42:04.830 where permutation wouldn’t work.

888 00:42:04.830 --> 00:42:07.080 If you just permute the cluster labels,

889 00:42:07.080 --> 00:42:09.750 the cells will look like the same cells, right?

890 00:42:09.750 --> 00:42:11.340 They’re still two clusters.

891 00:42:11.340 --> 00:42:12.690 But if you use our simulator,

892 00:42:12.690 --> 00:42:15.000 we could generate cells from one cluster

893 00:42:15.000 --> 00:42:18.720 that reflects the complete null, there’s no cluster.

894 00:42:18.720 --> 00:42:22.590 And the use of this can be shown in this example.

895 00:42:22.590 --> 00:42:24.450 There’s only one cluster,

896 00:42:24.450 --> 00:42:27.270 but if we use clustering algorithms,

897 00:42:27.270 --> 00:42:30.810 like these two choices, Seurat is a popular pipeline,

898 00:42:30.810 --> 00:42:33.990 Kmeans is the standard classical algorithm,

899 00:42:33.990 --> 00:42:38.100 using either to force the cells into two clusters,

900 00:42:38.100 --> 00:42:39.990 we are using gene expression data.

901 00:42:39.990 --> 00:42:43.560 So no wonder that if you look at a gene’s expression

902 00:42:43.560 --> 00:42:46.470 between the two clusters, you may call it DE,

903 00:42:46.470 --> 00:42:50.010 but that’s not interesting, since there’s no clusters.

904 00:42:50.010 --> 00:42:52.460 So if we use our scDesign3 to generate null data,

905 00:42:54.390 --> 00:42:58.313 in this case, null data should be very similar to real data.

906 00:42:58.313 --> 00:43:00.300 It still has only one cluster.

907 00:43:00.300 --> 00:43:03.960 Then, if we run Seurat or Kmeans,

908 00:43:03.960 --> 00:43:05.970 similarly, on null data,

909 00:43:05.970 --> 00:43:08.880 we would divide the cell in a similar way,

910 00:43:08.880 --> 00:43:12.150 and then, if you do a contrast of the two sets of results,

911 00:43:12.150 --> 00:43:13.800 you should see no big difference.

912 00:43:13.800 --> 00:43:16.200 That's the idea for controlling FDR.

913 00:43:16.200 --> 00:43:20.730 So indeed, in that example, if we're just naively wrong,

914 00:43:20.730 --> 00:43:25.110 the Seurat pipeline clustering followed by some tests

915 00:43:25.110 --> 00:43:27.750 like t, Wilcoxon, bimodal,

916 00:43:27.750 --> 00:43:30.480 yeah, you will see FDR is one.

917 00:43:30.480 --> 00:43:32.730 The reason is you keep finding D genes,

918 00:43:32.730 --> 00:43:34.200 even though there's no cluster.

919 00:43:34.200 --> 00:43:35.430 But using our approach,

920 00:43:35.430 --> 00:43:38.280 we could control the FDR reasonably well.

921 00:43:38.280 --> 00:43:41.520 So that's the predominant results for this purpose

922 00:43:41.520 --> 00:43:45.870 for this task, so that summarizes my talk today.

923 00:43:45.870 --> 00:43:48.150 And finally, I just want to make a few notes

924 00:43:48.150 --> 00:43:50.370 to give some messages.

925 00:43:50.370 --> 00:43:52.350 I talk about multiple testing,

926 00:43:52.350 --> 00:43:53.910 but in many scientific problems,

927 00:43:53.910 --> 00:43:57.240 I think the key is whether it should be formulated

928 00:43:57.240 --> 00:43:58.860 as a multiple testing problem.

929 00:43:58.860 --> 00:44:00.930 So actually, to address this question,

930 00:44:00.930 --> 00:44:02.910 I wrote a prospective article

931 00:44:02.910 --> 00:44:06.330 with my collaborator Xin Tong at USC.

932 00:44:06.330 --> 00:44:10.470 We try to clarify statistical hypothesis testing

933 00:44:10.470 --> 00:44:12.810 from machine learning binary classification.

934 00:44:12.810 --> 00:44:13.950 They seem similar

935 00:44:13.950 --> 00:44:17.010 because both would give you a binary decision, right?

936 00:44:17.010 --> 00:44:20.490 But I can say that testing is an inference problem,

937 00:44:20.490 --> 00:44:22.830 classification is a prediction problem.

938 00:44:22.830 --> 00:44:24.690 So if you really think about it,

939 00:44:24.690 --> 00:44:27.000 their fundamental concepts are different.

940 00:44:27.000 --> 00:44:30.900 So that's why we wrote this to really talk with biologists,

941 00:44:30.900 --> 00:44:34.530 for computational people who use this simultaneously.

942 00:44:34.530 --> 00:44:37.230 So if you're interested, you can check it out.

943 00:44:37.230 --> 00:44:39.780 And finally, I wanna say that,

944 00:44:39.780 --> 00:44:42.691 so if it's a multiple testing problem,

945 00:44:42.691 --> 00:44:47.580 I talked about three common causes of ill-posed P values,

946 00:44:47.580 --> 00:44:50.340 and I propose a solution, Clipper,

947 00:44:50.340 --> 00:44:54.630 for simplifying this problem by just using contrast scores,

948 00:44:54.630 --> 00:44:56.160 and then, set a cutoff.

949 00:44:56.160 --> 00:44:58.680 And the simulator, which we hope to be useful

950 00:44:58.680 --> 00:45:01.080 for the single cell and spatial omics field

951 00:45:01.080 --> 00:45:03.030 because this field is so popular,

952 00:45:03.030 --> 00:45:04.890 we have more than 1000 methods already.

953 00:45:04.890 --> 00:45:08.250 So benchmarking seems to be something quite necessary.

954 00:45:08.250 --> 00:45:10.650 Because if there's no benchmarking,

955 00:45:10.650 --> 00:45:13.710 then maybe new methods wouldn't have much of a chance

956 00:45:13.710 --> 00:45:16.170 because people may still use the older method

957 00:45:16.170 --> 00:45:18.030 that are better cited.

958 00:45:18.030 --> 00:45:22.950 Okay, so these are the papers related to my talk today.

959 00:45:22.950 --> 00:45:25.920 And so, finally, I want to say that,

960 00:45:25.920 --> 00:45:28.950 so if you're interested, you want to check them out,

961 00:45:28.950 --> 00:45:30.990 and let me know if you have any questions.

962 00:45:30.990 --> 00:45:32.550 So finally, I'll just say this,

963 00:45:32.550 --> 00:45:34.050 this is something quite interesting.

964 00:45:34.050 --> 00:45:36.930 It's another paper we just recently wrote,

965 00:45:36.930 --> 00:45:37.763 and I can say,

966 00:45:37.763 --> 00:45:40.290 you should be online in genome biology very soon.

967 00:45:40.290 --> 00:45:43.110 So we actually did this benchmark

968 00:45:43.110 --> 00:45:47.220 for the so called QTL analysis in genetics, right?

969 00:45:47.220 --> 00:45:49.770 Quantitative Trait Locus mapping.

970 00:45:49.770 --> 00:45:51.330 So in this analysis,

971 00:45:51.330 --> 00:45:55.320 a common procedure is to infer hidden variables

972 00:45:55.320 --> 00:45:57.360 from the data, like genes expression matrix,

973 00:45:57.360 --> 00:46:00.060 want to do hidden variable improvements.

974 00:46:00.060 --> 00:46:03.390 Besides the most part, (indistinct) has the classical PCA,

975 00:46:03.390 --> 00:46:06.690 several methods propose specific (indistinct).

976 00:46:06.690 --> 00:46:09.990 And my student Heather, actually gave her the full credit,

977 00:46:09.990 --> 00:46:12.930 she was so careful and she really wanted to understand

978 00:46:12.930 --> 00:46:14.400 the method before using it,

979 00:46:14.400 --> 00:46:16.560 then that lead to this project.

980 00:46:16.560 --> 00:46:19.350 She wants to see, huh, do I really see advantages

981 00:46:19.350 --> 00:46:22.290 of this new method even compared to PCA?

982 00:46:22.290 --> 00:46:23.880 But that's what she found, right?

983 00:46:23.880 --> 00:46:26.370 PCA still seems to be the most stable,

984 00:46:26.370 --> 00:46:29.610 robust, and also faster algorithm,

985 00:46:29.610 --> 00:46:32.400 but this is one of the reviewer's comments

986 00:46:32.400 --> 00:46:34.050 I wanna share with you.

987 00:46:34.050 --> 00:46:36.520 These results may come as a surprise to some,

988 00:46:36.520 --> 00:46:39.450 given the nearly un-contestable status

989 00:46:39.450 --> 00:46:42.060 that method A has achieved within the community.

990 00:46:42.060 --> 00:46:43.800 But sadly, they reflect the fact

991 00:46:43.800 --> 00:46:46.740 that computational biology methods can rise to fame

992 00:46:46.740 --> 00:46:50.280 almost by accident rather than by sound statistic arguments.

993 00:46:50.280 --> 00:46:51.570 So if you're interest,

994 00:46:51.570 --> 00:46:53.910 you can check out this paper, it's on bio archive.

995 00:46:53.910 --> 00:46:56.580 But anyway, I think it says how important it is

996 00:46:56.580 --> 00:46:59.880 for statisticians to convey our message, right?

997 00:46:59.880 --> 00:47:02.793 Why do we need statistical rigor, why does it matter?

998 00:47:03.720 --> 00:47:05.310 So for our students,

999 00:47:05.310 --> 00:47:07.776 if you want to know more about GAM and copulas,

1000 00:47:07.776 --> 00:47:09.630 there are two books I want to recommend.

1001 00:47:09.630 --> 00:47:12.270 So they're very good introductory textbooks,

1002 00:47:12.270 --> 00:47:14.490 so you can know the (indistinct).

1003 00:47:14.490 --> 00:47:18.900 Finally, I want to thank my collaborator at UC Irvine,

1004 00:47:18.900 --> 00:47:22.740 my students for all their tremendous work I talk about today

1005 00:47:22.740 --> 00:47:25.710 and also the funding agencies for giving us the support.

1006 00:47:25.710 --> 00:47:26.970 So thank you very much.

1007 00:47:36.226 --> 00:47:38.247 <v Attendee>A question?</v>

1008 00:47:38.247 --> 00:47:39.080 <v Jingyi>Yes.</v>

1009 00:47:39.080 --> 00:47:40.110 <v Attendee>So I was really curious</v>

1010 00:47:40.110 --> 00:47:44.850 about the analysis of like the large patient sample.

1011 00:47:44.850 --> 00:47:46.410 I know that there has in fact

1012 00:47:46.410 --> 00:47:47.980 been extensive discussion on it.

1013 00:47:47.980 --> 00:47:49.870 <v ->Yeah, yeah.</v> <v ->Which is</v>

1014 00:47:52.080 --> 00:47:54.690 interesting, to say the least, how it's gone down.

1015 00:47:54.690 --> 00:47:56.220 But I was kinda curious,

1016 00:47:56.220 --> 00:48:00.570 the way that it was presented here made me think about like,

1017 00:48:00.570 --> 00:48:04.803 apologies, if this is like a path that's already been tread,

1018 00:48:06.810 --> 00:48:10.890 so, yeah, the bar graph.

1019 00:48:10.890 --> 00:48:12.270 <v Jingyi>Yeah.</v>

1020 00:48:12.270 --> 00:48:15.750 <v Attendee>Yeah, so it sort of,</v>

1021 00:48:15.750 --> 00:48:18.870 it makes me wonder about the application

1022 00:48:18.870 --> 00:48:22.530 of the term false discovery in different contexts.

1023 00:48:22.530 --> 00:48:25.980 And taking patients, you can imagine,

1024 00:48:25.980 --> 00:48:29.490 there can be like unintended structure

1025 00:48:29.490 --> 00:48:32.370 within those populations.

1026 00:48:32.370 --> 00:48:34.230 And by (interference drowns out speaker) chance,

1027 00:48:34.230 --> 00:48:38.321 if there is 30,000 potential transcripts

1028 00:48:38.321 --> 00:48:40.710 that you're looking at, there might actually be,

1029 00:48:40.710 --> 00:48:43.890 between individuals who are not isogenic,

1030 00:48:43.890 --> 00:48:46.920 truly differentially expressed genes

1031 00:48:46.920 --> 00:48:50.220 between even permuted groups.

1032 00:48:50.220 --> 00:48:52.950 And so I'm wondering if there's a useful distinction

1033 00:48:52.950 --> 00:48:56.280 between a false discovery and a true,

1034 00:48:56.280 --> 00:48:58.173 but uninteresting discovery.

1035 00:49:00.240 --> 00:49:03.630 <v Jingyi>I think it depends on how you define truth.</v>

1036 00:49:03.630 --> 00:49:04.680 I think that's the key.

1037 00:49:04.680 --> 00:49:07.860 But what is the definition of D genes?

1038 00:49:07.860 --> 00:49:09.870 I wanna say, to be exact,

1039 00:49:09.870 --> 00:49:13.997 the definition of D genes in DESeq2,

1040 00:49:13.997 --> 00:49:17.550 edgeR, and that of Wilcoxon is different.

1041 00:49:17.550 --> 00:49:21.330 Because in Wilcoxon, the D gene is defined,

1042 00:49:21.330 --> 00:49:24.270 okay, if a gene, it has two distributions,

1043 00:49:24.270 --> 00:49:26.520 one under each condition,

1044 00:49:26.520 --> 00:49:29.280 and if I randomly take one observation

1045 00:49:29.280 --> 00:49:31.890 from each distribution from each condition,

1046 00:49:31.890 --> 00:49:34.650 is the chance that one is bigger than the other

1047 00:49:34.650 --> 00:49:35.850 equal to 0.5?

1048 00:49:35.850 --> 00:49:38.070 That's the Wilcoxon question.

1049 00:49:38.070 --> 00:49:41.520 While DESeq2 and edgeR, their D gene definition

1050 00:49:41.520 --> 00:49:45.090 is the negative binomial means are different.

1051 00:49:45.090 --> 00:49:48.060 But clearly, you can see, it only depends

1052 00:49:48.060 --> 00:49:51.360 on that negative binomial is a reasonable distribution,

1053 00:49:51.360 --> 00:49:52.193 that's the key.

1054 00:49:52.193 --> 00:49:53.026 So that's why in theory,

1055 00:49:53.026 --> 00:49:57.510 if negative binomial is no longer valid or reasonable,

1056 00:49:57.510 --> 00:50:00.090 then why should we define a D gene

1057 00:50:00.090 --> 00:50:02.550 based on negative binomial mean indifference?

1058 00:50:02.550 --> 00:50:05.550 I think that's kind of my answer to your question.

1059 00:50:05.550 --> 00:50:09.150 But the tricky thing about statistical inference

1060 00:50:09.150 --> 00:50:10.770 compared to supervised learning

1061 00:50:10.770 --> 00:50:14.370 is that we don't observe the truth, that's always the case.

1062 00:50:14.370 --> 00:50:16.110 So we're making a guess.

1063 00:50:16.110 --> 00:50:19.650 Frequentist people have one way to guess,

1064 00:50:19.650 --> 00:50:21.390 Poisson people have another way of guess.

1065 00:50:21.390 --> 00:50:23.700 And so one issue I've seen in the Twitter discussion

1066 00:50:23.700 --> 00:50:26.490 is that several people try to,

1067 00:50:26.490 --> 00:50:28.200 maybe not intentionally,

1068 00:50:28.200 --> 00:50:31.500 confuse frequentist concept with Poisson concept,

1069 00:50:31.500 --> 00:50:33.660 but they're not really comparable, right?

1070 00:50:33.660 --> 00:50:35.880 You cannot talk about them in the same ground.

1071 00:50:35.880 --> 00:50:39.270 That's a problem, and here, our criterion,

1072 00:50:39.270 --> 00:50:42.360 false discovery rate is a frequentist criteria,

1073 00:50:42.360 --> 00:50:43.590 it relies on P values, right?

1074 00:50:43.590 --> 00:50:46.410 So therefore, you cannot use Poisson arguments

1075 00:50:46.410 --> 00:50:49.350 to argue against such frequentist way.

1076 00:50:49.350 --> 00:50:51.630 Because you are doing frequentist, right?

1077 00:50:51.630 --> 00:50:53.700 But whether frequentist makes sense or not,

1078 00:50:53.700 --> 00:50:55.290 that's a different topic.

1079 00:50:55.290 --> 00:50:56.730 Hopefully, that answers your question.

1080 00:50:56.730 --> 00:50:58.520 <v ->Yeah, thank you.</v> <v ->Thank you.</v>

1081 00:51:00.870 --> 00:51:01.703 Yes. <v ->Hello,</v>

1082 00:51:01.703 --> 00:51:03.443 thank you much for your talk,

1083 00:51:03.443 --> 00:51:05.220 and I think that is very interesting.

1084 00:51:05.220 --> 00:51:09.410 However, I have a question on slide 26 actually..

1085 00:51:12.507 --> 00:51:15.007 It's about what you said that,

1086 00:51:17.180 --> 00:51:18.013 maybe 26.

1087 00:51:19.707 --> 00:51:22.695 <v Jingyi>26, okay, yeah.</v>

1088 00:51:22.695 --> 00:51:23.837 <v Attendee>Yeah, you said</v>

1089 00:51:23.837 --> 00:51:28.170 that like it is a multi-gene probabilistic model

1090 00:51:28.170 --> 00:51:29.640 for cell type.

1091 00:51:29.640 --> 00:51:33.300 However, I'm a little bit confused

1092 00:51:33.300 --> 00:51:35.593 about how you define the cell type.

1093 00:51:36.840 --> 00:51:40.410 But basically, from my own understandings,

1094 00:51:40.410 --> 00:51:44.103 that after you get, for example, the single cell rise data,

1095 00:51:45.210 --> 00:51:47.670 for example, you will use the route to get the cluster.

1096 00:51:47.670 --> 00:51:48.503 <v Jingyi>Yeah.</v>

1097 00:51:48.503 --> 00:51:51.996 <v Attendee>And you will annotate this cluster</v>

1098 00:51:51.996 --> 00:51:53.079 based on the-

1099 00:51:54.276 --> 00:51:56.305 <v ->Knowledge, yeah.</v> <v ->Gene.</v>

1100 00:51:56.305 --> 00:52:00.697 And then, if this model based on your

1101 00:52:04.090 --> 00:52:05.757 annotation of, okay.

1102 00:52:08.190 --> 00:52:09.840 <v Jingyi>Yeah, I see you point.</v>

1103 00:52:09.840 --> 00:52:12.660 Essentially, yeah, we need cell cluster to be pre-defined.

1104 00:52:12.660 --> 00:52:15.200 So if it's not reasonable, then, yes,

1105 00:52:15.200 --> 00:52:17.250 it will affect the results for sure.

1106 00:52:17.250 --> 00:52:19.710 Because the key is that you need to make sure

1107 00:52:19.710 --> 00:52:22.590 it is reasonable to assume a gene follows

1108 00:52:22.590 --> 00:52:26.070 one of the four distribution within a cluster, right?

1109 00:52:26.070 --> 00:52:27.900 So that's why there are methods

1110 00:52:27.900 --> 00:52:30.240 that try to refine clustering

1111 00:52:30.240 --> 00:52:33.660 by checking the negative binomial distribution.

1112 00:52:33.660 --> 00:52:35.970 So there are several research on that,

1113 00:52:35.970 --> 00:52:37.197 and they're trying to refine that.

1114 00:52:37.197 --> 00:52:40.050 But basically, we are sitting on those methods

1115 00:52:40.050 --> 00:52:42.600 to do the simulation, that's what we do.

1116 00:52:42.600 --> 00:52:46.287 But again, so that's why this is the problem with scDesign2,

1117 00:52:46.287 --> 00:52:49.950 but scDesign3 sort of tries to address this problem

1118 00:52:49.950 --> 00:52:51.960 by providing the BIC.

1119 00:52:51.960 --> 00:52:54.570 So if the input clusters are bad,

1120 00:52:54.570 --> 00:52:56.700 then you can see that in the BIC.

1121 00:52:56.700 --> 00:52:59.220 Because the likelihood will not be there, yeah.

1122 00:52:59.220 --> 00:53:00.623 <v Attendee>A similar question.</v>

1123 00:53:02.130 --> 00:53:03.789 I have another question

1124 00:53:03.789 --> 00:53:07.547 is that basically I assumed (indistinct) about

1125 00:53:08.632 --> 00:53:12.900 the experiments have duplicates,

1126 00:53:12.900 --> 00:53:16.300 however, in some situations,

1127 00:53:16.300 --> 00:53:20.130 maybe we do not have the replication.

1128 00:53:20.130 --> 00:53:24.103 But in this situation, how could we control the FDR,

1129 00:53:25.735 --> 00:53:27.724 if we do not have replicates,

1130 00:53:27.724 --> 00:53:29.970 then we cannot get the P value.

1131 00:53:29.970 --> 00:53:31.497 <v Jingyi>That's exactly the point of this talk.</v>

1132 00:53:31.497 --> 00:53:34.770 The only part that has replicates is the RNA-seq part.

1133 00:53:34.770 --> 00:53:37.200 The second part, that's the only part we have replicates.

1134 00:53:37.200 --> 00:53:39.037 In the first part, when we do the ChIP-seq,

1135 00:53:39.037 --> 00:53:41.880 it's just one replicate per condition, right?

1136 00:53:41.880 --> 00:53:44.700 That's why I said P value calculation would be helpful.

1137 00:53:44.700 --> 00:53:47.490 Right, so the reason we could control the FDR

1138 00:53:47.490 --> 00:53:49.350 without using P values

1139 00:53:49.350 --> 00:53:51.810 is just because we have many, many tests.

1140 00:53:51.810 --> 00:53:56.370 So that's why we're doing this large scale testing.

1141 00:53:56.370 --> 00:53:58.816 I think the idea, if you check it out,

1142 00:53:58.816 --> 00:54:03.780 Bran Efron has talked about it extensively in his book,

1143 00:54:03.780 --> 00:54:06.600 it's called, so his idea of Empirical Bayes

1144 00:54:06.600 --> 00:54:07.980 is very similar to this.

1145 00:54:07.980 --> 00:54:10.710 We try to borrow information across tests

1146 00:54:10.710 --> 00:54:12.505 to set a threshold.

1147 00:54:12.505 --> 00:54:14.758 Yeah, hopefully that answers your question.

1148 00:54:14.758 --> 00:54:15.591 Yeah?

1149 00:54:15.591 --> 00:54:19.758 (interference drowns out speaker)

1150 00:54:20.821 --> 00:54:22.657 Yeah, sounds good, thank you.

1151 00:54:22.657 --> 00:54:25.649 (interference drowns out speaker)

1152 00:54:25.649 --> 00:54:26.482 Thank you.