

# BaDGE (Bayesian model for Detecting Gene Environment interaction)

August 16, 2011

```
> library(BaDGE)
```

## Example Analysis

Load the data and print the first 5 rows.

```
> data(x, package = "BaDGE")
> x[1:5, ]
```

|   | ID        | Disease | Exposure | Gender | Age | SNP_1 | SNP_2 | SNP_3 | SNP_4 | SNP_5 | SNP_6 |
|---|-----------|---------|----------|--------|-----|-------|-------|-------|-------|-------|-------|
| 1 | subject_1 | 0       | 0        | 0      | 27  | 0     | 1     | 1     | 1     | 1     | 0     |
| 2 | subject_2 | 0       | 0        | 1      | 27  | 1     | 0     | 1     | 0     | 1     | 1     |
| 3 | subject_3 | 0       | 1        | 0      | 27  | 1     | 0     | 0     | 1     | 1     | 0     |
| 4 | subject_4 | 0       | 0        | 1      | 45  | 2     | 0     | 0     | 0     | 0     | 0     |
| 5 | subject_5 | 0       | 1        | 1      | 44  | 1     | 0     | 0     | 1     | 1     | 0     |

  

|   | SNP_7 | SNP_8 | SNP_9 | SNP_10 | SNP_11 | SNP_12 | SNP_13 | SNP_14 | SNP_15 |
|---|-------|-------|-------|--------|--------|--------|--------|--------|--------|
| 1 | 1     | 1     | 0     | 0      | 1      | 1      | 0      | 0      | 1      |
| 2 | 0     | 0     | 0     | 0      | 0      | 2      | 0      | 1      | 0      |
| 3 | 0     | 1     | 0     | 0      | 1      | 1      | 0      | 0      | 1      |
| 4 | 1     | 0     | 0     | 0      | 0      | 2      | 0      | 0      | 0      |
| 5 | 0     | 0     | 2     | 2      | 1      | 0      | 1      | 1      | 1      |

```
> dim(x)
```

```
[1] 2000 20
```

Next, we need to group subjects with the same multilocus genotype to be in the same group. The clustering algorithm treats each group as a unit.

```
> snps <- paste("SNP_", 1:15, sep = "")
> geno.mat <- x[, snps]
```

Function not run to save time. This will take a few minutes.

```
ret <- define.NB.geno(geno.mat)
```

Load the output from define.NB.geno.

```
> data(ret, package = "BaDGE")
```

Display the frequency counts for the groups.

```
> table(ret$grp.subj, exclude = NULL)
```

|     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |     |
|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  | 11  | 12  | 13  | 14  | 15  | 16  |
| 6   | 2   | 39  | 27  | 2   | 7   | 63  | 36  | 5   | 7   | 5   | 4   | 2   | 7   | 1   | 2   |
| 17  | 18  | 19  | 20  | 21  | 22  | 23  | 24  | 25  | 26  | 27  | 28  | 29  | 30  | 31  | 32  |
| 4   | 79  | 7   | 53  | 21  | 58  | 5   | 32  | 2   | 1   | 1   | 3   | 9   | 2   | 1   | 13  |
| 33  | 34  | 35  | 36  | 37  | 38  | 39  | 40  | 41  | 42  | 43  | 44  | 45  | 46  | 47  | 48  |
| 17  | 2   | 2   | 2   | 6   | 5   | 2   | 1   | 36  | 102 | 19  | 27  | 4   | 7   | 1   | 11  |
| 49  | 50  | 51  | 52  | 53  | 54  | 55  | 56  | 57  | 58  | 59  | 60  | 61  | 62  | 63  | 64  |
| 32  | 5   | 26  | 1   | 4   | 12  | 5   | 73  | 16  | 2   | 1   | 2   | 5   | 7   | 3   | 1   |
| 65  | 66  | 67  | 68  | 69  | 70  | 71  | 72  | 73  | 74  | 75  | 76  | 77  | 78  | 79  | 80  |
| 1   | 1   | 8   | 3   | 52  | 3   | 10  | 12  | 9   | 6   | 2   | 3   | 8   | 8   | 18  | 1   |
| 81  | 82  | 83  | 84  | 85  | 86  | 87  | 88  | 89  | 90  | 91  | 92  | 93  | 94  | 95  | 96  |
| 1   | 2   | 3   | 4   | 8   | 10  | 2   | 2   | 2   | 7   | 1   | 4   | 20  | 8   | 1   | 1   |
| 97  | 98  | 99  | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 |
| 3   | 1   | 1   | 1   | 1   | 7   | 1   | 4   | 1   | 3   | 4   | 1   | 1   | 5   | 6   | 1   |
| 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 |
| 16  | 7   | 8   | 7   | 5   | 2   | 5   | 1   | 2   | 2   | 6   | 5   | 2   | 1   | 5   | 1   |
| 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 |
| 3   | 9   | 2   | 19  | 3   | 2   | 7   | 1   | 1   | 2   | 4   | 18  | 7   | 4   | 7   | 2   |
| 145 | 146 | 147 | 148 | 149 | 150 | 151 | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 |
| 1   | 1   | 2   | 8   | 4   | 2   | 2   | 2   | 2   | 3   | 2   | 1   | 7   | 1   | 5   | 2   |
| 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 |
| 2   | 1   | 1   | 1   | 3   | 1   | 1   | 1   | 3   | 2   | 2   | 3   | 6   | 1   | 2   | 9   |
| 177 | 178 | 179 | 180 | 181 | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 |
| 4   | 1   | 3   | 2   | 4   | 2   | 2   | 32  | 1   | 3   | 2   | 4   | 1   | 2   | 2   | 2   |
| 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 |
| 1   | 1   | 5   | 1   | 4   | 2   | 1   | 2   | 1   | 1   | 15  | 1   | 2   | 1   | 4   | 4   |
| 209 | 210 | 211 | 212 | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 | 224 |
| 2   | 3   | 2   | 1   | 1   | 2   | 5   | 1   | 1   | 1   | 2   | 1   | 1   | 14  | 1   | 7   |
| 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 | 240 |
| 1   | 1   | 1   | 7   | 5   | 4   | 1   | 1   | 1   | 1   | 10  | 1   | 1   | 7   | 1   | 1   |
| 241 | 242 | 243 | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 | 256 |
| 1   | 2   | 3   | 1   | 1   | 1   | 19  | 1   | 1   | 2   | 1   | 4   | 1   | 5   | 1   | 5   |
| 257 | 258 | 259 | 260 | 261 | 262 | 263 | 264 | 265 | 266 | 267 | 268 | 269 | 270 | 271 | 272 |
| 5   | 4   | 1   | 1   | 1   | 2   | 1   | 1   | 4   | 1   | 1   | 2   | 1   | 2   | 1   | 1   |
| 273 | 274 | 275 | 276 | 277 | 278 | 279 | 280 | 281 | 282 | 283 | 284 | 285 | 286 | 287 | 288 |
| 5   | 1   | 2   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 1   | 4   | 1   | 1   |
| 289 | 290 | 291 | 292 | 293 | 294 | 295 | 296 | 297 | 298 | 299 | 300 | 301 | 302 | 303 | 304 |
| 1   | 1   | 1   | 1   | 2   | 2   | 7   | 3   | 2   | 2   | 3   | 1   | 1   | 1   | 6   | 1   |
| 305 | 306 | 307 | 308 | 309 | 310 | 311 | 312 | 313 | 314 | 315 | 316 | 317 | 318 | 319 | 320 |
| 1   | 3   | 2   | 1   | 1   | 2   | 1   | 1   | 2   | 1   | 2   | 2   | 1   | 1   | 1   | 2   |
| 321 | 322 | 323 | 324 | 325 | 326 | 327 | 328 | 329 | 330 | 331 | 332 | 333 | 334 | 335 | 336 |
| 1   | 4   | 2   | 2   | 1   | 5   | 2   | 5   | 3   | 1   | 2   | 1   | 3   | 1   | 1   | 2   |
| 337 | 338 | 339 | 340 | 341 | 342 | 343 | 344 | 345 | 346 | 347 | 348 | 349 | 350 | 351 | 352 |
| 3   | 2   | 1   | 3   | 1   | 1   | 1   | 1   | 4   | 1   | 1   | 1   | 1   | 2   | 3   | 2   |
| 353 | 354 | 355 | 356 | 357 | 358 | 359 | 360 | 361 | 362 | 363 | 364 | 365 | 366 | 367 | 368 |
| 1   | 2   | 3   | 2   | 1   | 2   | 1   | 1   | 3   | 1   | 1   | 3   | 1   | 1   | 1   | 1   |
| 369 | 370 | 371 | 372 | 373 | 374 | 375 | 376 | 377 | 378 | 379 | 380 | 381 | 382 | 383 | 384 |

```

1 2 1 1 3 2 1 2 2 3 1 2 1 1 1 3
385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400
3 1 1 2 1 1 1 2 2 1 2 1 1 2 2 1
401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432
1 1 1 1 1 1 2 1 1 1 1 2 2 1 1 1
433 434 435 436 437 <NA>
1 1 1 1 1 0

```

Add the group vector to the data frame x.

```
> x[, "Group"] <- ret$grp.subj
```

Define the folder where the output files will be written.

```
> outdir <- system.file("sampleData", package = "BaDGE")
```

Define the options for the badge function. We will use the similarity matrix created from the define.NB.geno function and run 1000000 iterations with output being written once for every 100 iterations. Other options include 2 clusters, 50 auxillary samples, alpha and beta parameters generated uniformly between -3 and 3. See the documentation for the list of all possible options.

```
> op <- list(sim.mat = ret$NB.mat, n_iter = 1e+06, n_sep_out = 100,
+           w_m = 50, k_max = 2, alpha_min = -3, alpha_max = 3, beta_min = -3,
+           beta_max = 3)
```

Call the main function, which is not run to save time. The estimated running time on a 2.8 GHz AMD Opteron 254 processor is 40 minutes.

```
badge(x, "Disease", "Exposure", "Group", outdir, op=op)
```

The output files contain samples generated from the MCMC algorithm. We have a post-processing function that can be used to generate ssummary plots. It also outputs summary statistics for each subject (see the help documentation for more details). Define the options list for processing the results. Since we ran the badge function with n\_iter=1000000 and n\_sep\_out=100, each output file will contain 1000000/100 = 10000 rows. Let the first 1000 rows (100,000 iterations) be the burn in period.

```
> op$M1 <- 1000
```

Process the results

```
> ret <- post_badge(geno.mat, x, "Disease", "Exposure", "Group",
+                 outdir, op = op)
> names(ret)
```

```
[1] "dic"           "subj.assign"   "alpha.med.odds" "beta.med.odds"
[5] "pc.mat"
```

## Session Information

```
> sessionInfo()
```

```
R version 2.11.0 (2010-04-22)  
x86_64-pc-mingw32
```

```
locale:
```

```
[1] LC_COLLATE=English_United States.1252  
[2] LC_CTYPE=English_United States.1252  
[3] LC_MONETARY=English_United States.1252  
[4] LC_NUMERIC=C  
[5] LC_TIME=English_United States.1252
```

```
attached base packages:
```

```
[1] tools      stats      graphics  grDevices  utils      datasets  methods  
[8] base
```

```
other attached packages:
```

```
[1] BaDGE_1.0.0    cluster_1.12.3  fields_6.3     spam_0.23-0
```