

APPENDIX

Table A.1. Estimation of Molecular Weights of EnvZ Fragments

Protein	Calculated M.W. (kDa)	Measured M.W. (kDa)	Form	
			Gel Filtration	Light Scattering
Cytoplasmic Region (223 - 450)	24.9	not done	57 - 59	Dimer
Domain A (223 - 289)	7.6	15	19 - 21	Dimer
Domain B (290 - 450)	17.6	19	25 - 28	Monomer

INTRAMOLECULAR NOES IN DOMAIN B

assign (resid 292 and name HA) (resid 292 and name HB#) 0.0 1.0 3.9
assign (resid 292 and name HA) (resid 292 and name HC#) 0.0 1.0 3.9
assign (resid 293 and name HA) (resid 293 and name HB#) 0.0 1.0 3.9
assign (resid 293 and name HA) (resid 293 and name HC#) 0.0 1.0 3.9
assign (resid 293 and name HA) (resid 293 and name HN) 0.0 0.0 5.0
assign (resid 293 and name HB#) (resid 293 and name HN) 0.0 0.6 4.1
assign (resid 293 and name HG#) (resid 293 and name HN) 0.0 1.0 6.0
assign (resid 294 and name HA) (resid 294 and name HB#) 0.0 1.0 4.5
assign (resid 294 and name HA) (resid 294 and name HC#) 0.0 1.0 3.9
assign (resid 294 and name HB#) (resid 294 and name HN) 0.0 0.6 4.1
assign (resid 294 and name HG#) (resid 294 and name HN) 0.0 1.0 6.0
assign (resid 295 and name HA) (resid 295 and name HB#) 0.0 1.0 3.9
assign (resid 295 and name HA) (resid 295 and name HC#) 0.0 1.0 3.9
assign (resid 295 and name HA) (resid 295 and name HE#) 0.0 1.0 6.0
assign (resid 295 and name HA) (resid 295 and name HN) 0.0 0.6 3.1
assign (resid 295 and name HB#) (resid 295 and name HN) 0.0 1.0 4.5
assign (resid 295 and name HG#) (resid 295 and name HN) 0.0 2.0 5.5
assign (resid 296 and name HA) (resid 296 and name HB#) 0.0 1.0 3.9
assign (resid 296 and name HA) (resid 296 and name HE#) 0.0 1.5 4.4
assign (resid 296 and name HA) (resid 296 and name HC#) 0.0 1.0 3.9
assign (resid 296 and name HB#) (resid 296 and name HC#) 0.0 1.0 3.9
assign (resid 296 and name HB#) (resid 296 and name HE#) 0.0 2.5 5.0
assign (resid 296 and name HB#) (resid 296 and name HN) 0.0 0.6 3.1
assign (resid 296 and name HE#) (resid 296 and name HN) 0.0 2.5 5.0
assign (resid 296 and name HG#) (resid 296 and name HN) 0.0 1.0 3.9
assign (resid 297 and name HA) (resid 297 and name HB#) 0.0 1.0 3.9
assign (resid 297 and name HA) (resid 297 and name HE#) 0.0 1.0 3.9
assign (resid 297 and name HA) (resid 297 and name HC#) 0.0 1.0 6.0
assign (resid 297 and name HB#) (resid 297 and name HN) 0.0 0.0 3.5
assign (resid 297 and name HB#) (resid 297 and name HN) 0.0 0.6 3.5
assign (resid 297 and name HG#) (resid 297 and name HN) 0.0 1.0 3.9
assign (resid 298 and name HA) (resid 298 and name HE#) 0.0 1.5 4.4
assign (resid 298 and name HA) (resid 298 and name HG#) 0.0 1.0 4.5
assign (resid 298 and name HA) (resid 298 and name HN) 0.0 0.0 2.9
assign (resid 298 and name HB#) (resid 298 and name HN) 0.0 0.6 3.1
assign (resid 298 and name HE#) (resid 298 and name HN) 0.0 2.5 5.0
assign (resid 298 and name HG#) (resid 298 and name HN) 0.0 1.0 3.9
assign (resid 298 and name HC#) (resid 298 and name HN) 0.0 1.0 3.9
assign (resid 299 and name HA) (resid 299 and name HB#) 0.0 1.0 4.0
assign (resid 299 and name HA) (resid 299 and name HN) 0.0 0.0 2.9
assign (resid 299 and name HB#) (resid 299 and name HN) 0.0 1.1 4.0
assign (resid 300 and name HA) (resid 300 and name HN) 0.0 0.0 2.9
assign (resid 300 and name HB#) (resid 300 and name HN) 0.0 0.6 3.5
assign (resid 300 and name HE#) (resid 300 and name HN) 0.0 1.5 5.0
assign (resid 301 and name HA) (resid 301 and name HD1#) 0.0 1.5 4.4
assign (resid 301 and name HA) (resid 301 and name HD2#) 0.0 1.5 4.4
assign (resid 301 and name HB#) (resid 301 and name HN) 0.0 0.0 5.0
assign (resid 301 and name HD1#) (resid 301 and name HN) 0.0 1.5 5.0
assign (resid 301 and name HD2#) (resid 301 and name HN) 0.0 1.5 5.0
assign (resid 301 and name HG) (resid 301 and name HN) 0.0 0.0 3.5
assign (resid 302 and name HA) (resid 302 and name HB#) 0.0 1.0 4.5
assign (resid 302 and name HA) (resid 302 and name HN) 0.0 0.0 3.5
assign (resid 302 and name HB#) (resid 302 and name HN) 0.0 0.0 3.5
assign (resid 302 and name HB#) (resid 302 and name HD2#) 0.0 2.0 5.5
assign (resid 302 and name HB#) (resid 302 and name HN) 0.0 0.6 3.5
assign (resid 303 and name HA) (resid 303 and name HN) 0.0 1.0 6.0
assign (resid 303 and name HB#) (resid 303 and name HN) 0.0 0.0 2.9
assign (resid 303 and name HB#) (resid 303 and name HN) 0.0 1.1 3.6
assign (resid 304 and name HA) (resid 304 and name HB) 0.0 0.0 3.5
assign (resid 304 and name HA) (resid 304 and name HB#) 0.0 1.5 4.0
assign (resid 304 and name HA) (resid 304 and name HG1#) 0.0 1.5 4.0
assign (resid 304 and name HA) (resid 304 and name HG2#) 0.0 1.5 4.0
assign (resid 304 and name HA) (resid 304 and name HN) 0.0 0.0 2.9
assign (resid 304 and name HB) (resid 304 and name HN) 0.0 0.0 2.9
assign (resid 304 and name HB#) (resid 304 and name HN) 0.0 1.5 4.4
assign (resid 304 and name HB#) (resid 304 and name HN) 0.0 1.5 4.4
assign (resid 304 and name HG1#) (resid 304 and name HN) 0.0 1.5 4.4
assign (resid 304 and name HG2#) (resid 304 and name HN) 0.0 1.5 4.4
assign (resid 305 and name HA) (resid 305 and name HB#) 0.0 1.0 6.0
assign (resid 305 and name HA) (resid 305 and name HD1#) 0.0 1.5 4.4
assign (resid 305 and name HA) (resid 305 and name HD2#) 0.0 1.5 4.4

Table A.2. (continued)

assign (resid 330 and name HG2#)	(resid 330 and name HN)	0.0 1.5 5.0
assign (resid 330 and name HA)	(resid 330 and name HB#)	0.0 1.5 4.4
assign (resid 331 and name HA)	(resid 331 and name HB#)	0.0 1.0 4.5
assign (resid 331 and name HA)	(resid 331 and name HD#)	0.0 1.0 4.5
assign (resid 331 and name HA)	(resid 331 and name HG#)	0.0 1.0 4.5
assign (resid 331 and name HA)	(resid 331 and name HG#)	0.0 1.0 6.0
assign (resid 331 and name HA)	(resid 331 and name HE#)	0.0 2.0 4.9
assign (resid 331 and name HB#)	(resid 331 and name HN)	0.0 0.6 3.5
assign (resid 331 and name HB#)	(resid 331 and name HN)	0.0 1.0 3.9
assign (resid 331 and name HE#)	(resid 331 and name HG#)	0.0 2.0 4.5
assign (resid 331 and name HE#)	(resid 331 and name HN)	0.0 1.0 7.0
assign (resid 331 and name HB#)	(resid 331 and name HN)	0.0 1.0 4.5
assign (resid 332 and name HA)	(resid 332 and name HE#)	0.0 1.5 5.0
assign (resid 332 and name HA)	(resid 332 and name HN)	0.0 0.0 5.0
assign (resid 332 and name HB#)	(resid 332 and name HN)	0.0 0.6 4.1
assign (resid 333 and name HA)	(resid 333 and name HB#)	0.0 1.0 4.5
assign (resid 333 and name HA)	(resid 333 and name HD2)	0.0 0.0 2.9
assign (resid 333 and name HA)	(resid 333 and name HD)	0.0 0.0 5.0
assign (resid 333 and name HB#)	(resid 333 and name HD2)	0.0 1.0 6.0
assign (resid 333 and name HB#)	(resid 333 and name HN)	0.0 0.6 5.6
assign (resid 334 and name HA)	(resid 334 and name HB#)	0.0 1.0 4.5
assign (resid 335 and name HA)	(resid 335 and name HB#)	0.0 1.0 3.9
assign (resid 335 and name HA)	(resid 335 and name HD#)	0.0 1.7 4.2
assign (resid 335 and name HA)	(resid 335 and name HG)	0.0 0.0 3.5
assign (resid 335 and name HA)	(resid 335 and name HN)	0.0 0.0 2.9
assign (resid 335 and name HB#)	(resid 335 and name HD#)	0.0 3.4 5.9
assign (resid 335 and name HB#)	(resid 335 and name HN)	0.0 0.6 3.5
assign (resid 336 and name HA)	(resid 336 and name HB#)	0.0 1.7 4.6
assign (resid 336 and name HA)	(resid 336 and name HN)	0.0 1.0 4.5
assign (resid 336 and name HB#)	(resid 336 and name HN)	0.0 0.0 3.5
assign (resid 337 and name HA)	(resid 337 and name HB#)	0.0 0.6 5.6
assign (resid 337 and name HA)	(resid 337 and name HD1#)	0.0 1.5 4.4
assign (resid 337 and name HA)	(resid 337 and name HD2#)	0.0 1.5 4.4
assign (resid 337 and name HA)	(resid 337 and name HN)	0.0 0.0 5.0
assign (resid 337 and name HB)	(resid 337 and name HN)	0.0 0.0 2.9
assign (resid 337 and name HD1#)	(resid 337 and name HG2#)	0.0 3.0 5.5
assign (resid 337 and name HD1#)	(resid 337 and name HN)	0.0 1.5 5.0
assign (resid 337 and name HD2#)	(resid 337 and name HN)	0.0 1.5 5.0
assign (resid 338 and name HA)	(resid 338 and name HB#)	0.0 1.0 4.5
assign (resid 338 and name HA)	(resid 338 and name HD#)	0.0 1.0 6.0
assign (resid 338 and name HA)	(resid 338 and name HG#)	0.0 1.0 4.5
assign (resid 338 and name HB#)	(resid 338 and name HN)	0.0 0.0 2.9
assign (resid 338 and name HB#)	(resid 338 and name HN)	0.0 0.6 3.5
assign (resid 338 and name HE#)	(resid 338 and name HG#)	0.0 2.0 7.0
assign (resid 338 and name HG#)	(resid 338 and name HN)	0.0 1.0 7.0
assign (resid 339 and name HA)	(resid 339 and name HD#)	0.0 1.0 3.9
assign (resid 339 and name HA)	(resid 339 and name HN)	0.0 0.0 2.9
assign (resid 339 and name HA)	(resid 339 and name HN)	0.0 1.0 6.0
assign (resid 340 and name HA)	(resid 340 and name HN)	0.0 0.0 2.9
assign (resid 340 and name HB#)	(resid 340 and name HN)	0.0 1.1 4.0
assign (resid 341 and name HA)	(resid 341 and name HB)	0.0 0.0 3.5
assign (resid 341 and name HA)	(resid 341 and name HG1#)	0.0 1.5 4.4
assign (resid 341 and name HA)	(resid 341 and name HG2#)	0.0 1.5 4.4
assign (resid 341 and name HA)	(resid 341 and name HN)	0.0 0.0 2.9
assign (resid 341 and name HA)	(resid 341 and name HN)	0.0 0.0 2.9
assign (resid 341 and name HB#)	(resid 341 and name HN)	0.0 1.5 4.4
assign (resid 341 and name HB#)	(resid 341 and name HN)	0.0 1.5 4.4
assign (resid 342 and name HA)	(resid 342 and name HN)	0.0 0.0 2.9

Table A.2. (continued)

assign (resid 342 and name HB#)	(resid 342 and name HN)	0.0 1.1 3.6
assign (resid 343 and name HA)	(resid 343 and name HB#)	0.0 1.0 6.0
assign (resid 343 and name HA)	(resid 343 and name HN)	0.0 0.0 3.5
assign (resid 343 and name HB#)	(resid 343 and name HN)	0.0 0.6 4.1
assign (resid 344 and name HA)	(resid 344 and name HG#)	0.0 1.7 4.6
assign (resid 345 and name HA)	(resid 345 and name HN)	0.0 0.0 5.0
assign (resid 345 and name HG#)	(resid 345 and name HN)	0.0 1.7 5.2
assign (resid 346 and name HA)	(resid 346 and name HB)	0.0 0.0 3.5
assign (resid 346 and name HA)	(resid 346 and name HG#)	0.0 1.7 4.6
assign (resid 347 and name HA)	(resid 347 and name HB#)	0.0 1.0 4.5
assign (resid 348 and name HA)	(resid 348 and name HN)	0.0 0.0 5.0
assign (resid 348 and name HB#)	(resid 348 and name HN)	0.0 1.1 6.1
assign (resid 349 and name HA)	(resid 349 and name HN)	0.0 0.0 2.9
assign (resid 349 and name HB#)	(resid 349 and name HN)	0.0 1.1 3.6
assign (resid 350 and name HA)	(resid 350 and name HD#)	0.0 1.0 4.5
assign (resid 350 and name HA)	(resid 350 and name HG#)	0.0 1.0 4.5
assign (resid 350 and name HA)	(resid 350 and name HN)	0.0 0.0 3.5
assign (resid 350 and name HD#)	(resid 350 and name HN)	0.0 1.0 6.0
assign (resid 350 and name HA)	(resid 350 and name HB#)	0.0 1.0 3.9
assign (resid 353 and name HA)	(resid 353 and name HB#)	0.0 1.0 3.9
assign (resid 353 and name HB#)	(resid 353 and name HD2#)	0.0 1.0 6.0
assign (resid 353 and name HB#)	(resid 353 and name HD2#)	0.0 2.0 5.5
assign (resid 354 and name HA#)	(resid 354 and name HN)	0.0 0.6 4.1
assign (resid 355 and name HA)	(resid 355 and name HB#)	0.0 1.0 3.9
assign (resid 355 and name HA)	(resid 355 and name HD1)	0.0 0.0 3.5
assign (resid 355 and name HA)	(resid 355 and name HN)	0.0 0.0 5.0
assign (resid 355 and name HB#)	(resid 355 and name HD1)	0.0 1.0 4.5
assign (resid 355 and name HB#)	(resid 355 and name HE1)	0.0 1.0 6.0
assign (resid 355 and name HB#)	(resid 355 and name HN)	0.0 0.6 5.6
assign (resid 356 and name HA)	(resid 356 and name HN)	0.0 0.0 6.0
assign (resid 356 and name HA)	(resid 356 and name HG2#)	0.0 1.5 4.4
assign (resid 356 and name HA)	(resid 356 and name HN)	0.0 0.0 3.5
assign (resid 356 and name HB)	(resid 356 and name HN)	0.0 0.0 2.9
assign (resid 356 and name HB)	(resid 356 and name HN)	0.0 1.5 4.4
assign (resid 357 and name HA)	(resid 357 and name HN)	0.0 0.0 5.0
assign (resid 358 and name HA)	(resid 358 and name HB)	0.0 0.0 2.9
assign (resid 358 and name HA)	(resid 358 and name HG1#)	0.0 1.5 4.4
assign (resid 358 and name HA)	(resid 358 and name HG2#)	0.0 1.5 4.4
assign (resid 358 and name HB)	(resid 358 and name HN)	0.0 0.0 3.5
assign (resid 358 and name HG2#)	(resid 358 and name HN)	0.0 1.5 4.4
assign (resid 359 and name HA)	(resid 359 and name HB#)	0.0 1.0 4.5
assign (resid 359 and name HA)	(resid 359 and name HN)	0.0 0.0 5.0
assign (resid 359 and name HB#)	(resid 359 and name HN)	0.0 0.6 5.6
assign (resid 360 and name HA)	(resid 360 and name HN)	0.0 0.0 2.9
assign (resid 361 and name HA#)	(resid 361 and name HN)	0.0 0.6 4.1
assign (resid 362 and name HA)	(resid 362 and name HB)	0.0 0.0 2.9
assign (resid 362 and name HA)	(resid 362 and name HG2#)	0.0 1.5 4.4
assign (resid 362 and name HB)	(resid 362 and name HN)	0.0 0.0 3.5
assign (resid 362 and name HB)	(resid 362 and name HN)	0.0 0.0 5.0
assign (resid 363 and name HA)	(resid 363 and name HN)	0.0 1.5 4.4
assign (resid 363 and name HA)	(resid 363 and name HG#)	0.0 1.0 3.9
assign (resid 363 and name HG#)	(resid 363 and name HN)	0.0 1.0 4.5
assign (resid 364 and name HA)	(resid 364 and name HB#)	0.0 1.0 3.9
assign (resid 364 and name HA)	(resid 364 and name HG#)	0.0 1.0 4.5
assign (resid 365 and name HA)	(resid 365 and name HB#)	0.0 1.0 3.9
assign (resid 365 and name HB#)	(resid 365 and name HD2#)	0.0 2.0 5.5
assign (resid 366 and name HA)	(resid 366 and name HB#)	0.0 0.6 5.6
assign (resid 366 and name HA)	(resid 366 and name HB#)	0.0 1.0 4.5
assign (resid 366 and name HA)	(resid 366 and name HG#)	0.0 1.0 6.0
assign (resid 366 and name HA)	(resid 366 and name HN)	0.0 0.0 3.5

Table A.2. (continued)

Table with 4 columns: assign (resid), name HA, name HB#, and name HN. Rows 316-339. Data includes resid values and names like HB#, HD#, HG1#, HG2#, HD1#, HD2#.

Table A.2. (continued)

Table with 4 columns: assign (resid), name HB#, name HD#, and name HN. Rows 329-344. Data includes resid values and names like HB#, HD#, HG1#, HG2#, HD1#, HD2#.

Table A.3. Dihedral Angle Constraints Used for Structure Calculation of Domain B

i	dihedral angle constraints from HMQCJ	weight
assign	(resid 293 and name c)(resid 294 and name n)	-120.000
assign	(resid 294 and name c)(resid 294 and name c)	50.0000 2
assign	(resid 296 and name c)(resid 297 and name n)	-120.000
assign	(resid 297 and name c)(resid 297 and name c)	50.0000 2
assign	(resid 302 and name c)(resid 303 and name n)	-60.000
assign	(resid 303 and name c)(resid 303 and name c)	30.0000 2
assign	(resid 304 and name c)(resid 304 and name n)	-60.000
assign	(resid 304 and name c)(resid 305 and name n)	30.0000 2
assign	(resid 305 and name c)(resid 305 and name c)	-60.000
assign	(resid 307 and name c)(resid 307 and name n)	-60.000
assign	(resid 308 and name c)(resid 308 and name n)	30.0000 2
assign	(resid 309 and name c)(resid 309 and name c)	-60.000
assign	(resid 310 and name c)(resid 310 and name n)	-60.000
assign	(resid 311 and name c)(resid 311 and name n)	30.0000 2
assign	(resid 316 and name c)(resid 317 and name n)	-120.000
assign	(resid 317 and name c)(resid 317 and name c)	50.0000 2
assign	(resid 318 and name c)(resid 318 and name n)	-120.000
assign	(resid 319 and name c)(resid 319 and name c)	30.0000 2
assign	(resid 320 and name c)(resid 320 and name n)	-120.000
assign	(resid 321 and name c)(resid 321 and name n)	30.0000 2
assign	(resid 322 and name c)(resid 322 and name n)	-120.000
assign	(resid 327 and name c)(resid 328 and name n)	-120.000
assign	(resid 328 and name c)(resid 328 and name c)	50.0000 2
assign	(resid 330 and name c)(resid 330 and name n)	-120.000
assign	(resid 334 and name c)(resid 335 and name n)	-60.000
assign	(resid 335 and name c)(resid 335 and name c)	30.0000 2
assign	(resid 338 and name c)(resid 338 and name n)	-60.000
assign	(resid 339 and name c)(resid 340 and name n)	-60.000
assign	(resid 340 and name c)(resid 340 and name c)	30.0000 2
assign	(resid 341 and name c)(resid 341 and name n)	-60.000
assign	(resid 355 and name c)(resid 356 and name n)	-120.000
assign	(resid 356 and name c)(resid 356 and name c)	30.0000 2
assign	(resid 357 and name c)(resid 357 and name n)	-120.000
assign	(resid 358 and name c)(resid 358 and name n)	30.0000 2
assign	(resid 366 and name c)(resid 367 and name n)	-120.000
assign	(resid 367 and name c)(resid 367 and name c)	30.0000 2
assign	(resid 368 and name c)(resid 369 and name n)	-120.000
assign	(resid 369 and name c)(resid 369 and name c)	30.0000 2
assign	(resid 370 and name c)(resid 370 and name n)	-120.000
assign	(resid 371 and name c)(resid 372 and name n)	-120.000
assign	(resid 372 and name c)(resid 372 and name c)	30.0000 2
assign	(resid 373 and name c)(resid 374 and name n)	-120.000
assign	(resid 374 and name c)(resid 374 and name c)	50.0000 2

Table A.2. (continued)

assign	(resid 369 and name c)	0.0 0.0 2.3
assign	(resid 419 and name o)	0.0 0.0 3.3
assign	(resid 419 and name o)	0.0 0.0 2.3
assign	(resid 436 and name o)	0.0 0.0 3.3
assign	(resid 436 and name o)	0.0 0.0 2.3
assign	(resid 330 and name o)	0.0 0.0 3.3
assign	(resid 330 and name o)	0.0 0.0 2.3
assign	(resid 356 and name o)	0.0 0.0 3.3
assign	(resid 356 and name o)	0.0 0.0 2.3
assign	(resid 358 and name o)	0.0 0.0 3.3
assign	(resid 358 and name o)	0.0 0.0 2.3
assign	(resid 360 and name o)	0.0 0.0 3.3
assign	(resid 360 and name o)	0.0 0.0 2.3
assign	(resid 297 and name o)	0.0 0.0 3.3
assign	(resid 297 and name o)	0.0 0.0 2.3
assign	(resid 318 and name o)	0.0 0.0 3.3
assign	(resid 318 and name o)	0.0 0.0 2.3
assign	(resid 355 and name o)	0.0 0.0 3.3
assign	(resid 355 and name o)	0.0 0.0 2.3
assign	(resid 431 and name o)	0.0 0.0 3.3
assign	(resid 431 and name o)	0.0 0.0 2.3
assign	(resid 423 and name o)	0.0 0.0 3.3
assign	(resid 423 and name o)	0.0 0.0 2.3
assign	(resid 423 and name o)	0.0 0.0 3.3
assign	(resid 423 and name o)	0.0 0.0 2.3
assign	(resid 366 and name o)	0.0 0.0 3.3
assign	(resid 366 and name o)	0.0 0.0 2.3
assign	(resid 366 and name o)	0.0 0.0 3.3
assign	(resid 366 and name o)	0.0 0.0 2.3
assign	(resid 435 and name o)	0.0 0.0 3.3
assign	(resid 435 and name o)	0.0 0.0 2.3
assign	(resid 429 and name o)	0.0 0.0 3.3
assign	(resid 429 and name o)	0.0 0.0 2.3
assign	(resid 425 and name o)	0.0 0.0 3.3
assign	(resid 425 and name o)	0.0 0.0 2.3

Table A.4. (continued)

Table with 4 columns: assign, (segid b and resid), name HN, and name HB#. Rows list sequential assignments for various segment and residue identifiers.

Table A.4. (continued)

Table with 4 columns: assign, (segid b and resid), name HA, and name HD1#. Rows list sequential assignments for various segment and residue identifiers.

Table A.S. (continued)

assign (segid b and resid 249 and name n) (segid b and resid 249 and name ca)	30.0	2
assign (segid b and resid 249 and name c) (segid b and resid 250 and name n) 1 -50.0	30.0	2
assign (segid b and resid 250 and name n) (segid b and resid 250 and name ca)	30.0	2
assign (segid b and resid 251 and name n) (segid b and resid 251 and name ca)	30.0	2
assign (segid b and resid 251 and name c) (segid b and resid 252 and name n) 1 -50.0	30.0	2
assign (segid b and resid 252 and name n) (segid b and resid 253 and name ca)	30.0	2
assign (segid b and resid 253 and name c) (segid b and resid 254 and name n) 1 -50.0	30.0	2
assign (segid b and resid 254 and name n) (segid b and resid 254 and name ca)	30.0	2
assign (segid b and resid 254 and name c) (segid b and resid 255 and name n) 1 -50.0	30.0	2
assign (segid b and resid 255 and name n) (segid b and resid 255 and name ca)	30.0	2
assign (segid b and resid 255 and name c) (segid b and resid 256 and name n) 1 -50.0	30.0	2
assign (segid b and resid 256 and name n) (segid b and resid 256 and name ca)	30.0	2
assign (segid b and resid 256 and name c) (segid b and resid 257 and name n) 1 -50.0	30.0	2
assign (segid b and resid 257 and name n) (segid b and resid 257 and name ca)	30.0	2
assign (segid b and resid 257 and name c) (segid b and resid 258 and name n) 1 -50.0	30.0	2
assign (segid b and resid 258 and name n) (segid b and resid 264 and name ca)	30.0	2
assign (segid b and resid 264 and name c) (segid b and resid 265 and name n) 1 -50.0	30.0	2
assign (segid b and resid 265 and name n) (segid b and resid 265 and name ca)	30.0	2
assign (segid b and resid 265 and name c) (segid b and resid 266 and name n) 1 -50.0	30.0	2
assign (segid b and resid 266 and name n) (segid b and resid 266 and name ca)	30.0	2
assign (segid b and resid 266 and name c) (segid b and resid 267 and name n) 1 -50.0	30.0	2
assign (segid b and resid 267 and name n) (segid b and resid 267 and name ca)	30.0	2
assign (segid b and resid 267 and name c) (segid b and resid 268 and name n) 1 -50.0	30.0	2
assign (segid b and resid 268 and name n) (segid b and resid 268 and name ca)	30.0	2
assign (segid b and resid 268 and name c) (segid b and resid 269 and name n) 1 -50.0	30.0	2
assign (segid b and resid 269 and name n) (segid b and resid 269 and name ca)	30.0	2
assign (segid b and resid 269 and name c) (segid b and resid 270 and name n) 1 -50.0	30.0	2
assign (segid b and resid 270 and name n) (segid b and resid 270 and name ca)	30.0	2
assign (segid b and resid 270 and name c) (segid b and resid 271 and name n) 1 -50.0	30.0	2
assign (segid b and resid 271 and name n) (segid b and resid 271 and name ca)	30.0	2
assign (segid b and resid 271 and name c) (segid b and resid 272 and name n) 1 -50.0	30.0	2
assign (segid b and resid 272 and name n) (segid b and resid 272 and name ca)	30.0	2
assign (segid b and resid 272 and name c) (segid b and resid 273 and name n) 1 -50.0	30.0	2
assign (segid b and resid 273 and name n) (segid b and resid 273 and name ca)	30.0	2
assign (segid b and resid 273 and name c) (segid b and resid 274 and name n) 1 -50.0	30.0	2
assign (segid b and resid 274 and name n) (segid b and resid 274 and name ca)	30.0	2
assign (segid b and resid 274 and name c) (segid b and resid 275 and name n) 1 -50.0	30.0	2
assign (segid b and resid 275 and name n) (segid b and resid 275 and name ca)	30.0	2
assign (segid b and resid 275 and name c) (segid b and resid 276 and name n) 1 -50.0	30.0	2
assign (segid b and resid 276 and name n) (segid b and resid 276 and name ca)	30.0	2
assign (segid b and resid 276 and name c) (segid b and resid 277 and name n) 1 -50.0	30.0	2
assign (segid b and resid 277 and name n) (segid b and resid 277 and name ca)	30.0	2
assign (segid b and resid 277 and name c) (segid b and resid 278 and name n) 1 -50.0	30.0	2
assign (segid b and resid 278 and name n) (segid b and resid 278 and name ca)	30.0	2
assign (segid b and resid 278 and name c) (segid b and resid 279 and name n) 1 -50.0	30.0	2
assign (segid b and resid 279 and name n) (segid b and resid 279 and name ca)	30.0	2
assign (segid b and resid 279 and name c) (segid b and resid 280 and name n) 1 -50.0	30.0	2
assign (segid b and resid 280 and name n) (segid b and resid 280 and name ca)	30.0	2
assign (segid b and resid 280 and name c) (segid b and resid 281 and name n) 1 -50.0	30.0	2
assign (segid b and resid 281 and name n) (segid b and resid 281 and name ca)	30.0	2
assign (segid b and resid 281 and name c) (segid b and resid 282 and name n) 1 -50.0	30.0	2
assign (segid b and resid 282 and name n) (segid b and resid 282 and name ca)	30.0	2
assign (segid b and resid 282 and name c) (segid b and resid 283 and name n) 1 -50.0	30.0	2
assign (segid b and resid 283 and name n) (segid b and resid 283 and name ca)	30.0	2
assign (segid b and resid 283 and name c) (segid b and resid 284 and name n) 1 -50.0	30.0	2
assign (segid b and resid 284 and name n) (segid b and resid 284 and name ca)	30.0	2
assign (segid b and resid 284 and name c) (segid b and resid 285 and name n) 1 -50.0	30.0	2
assign (segid b and resid 285 and name n) (segid b and resid 285 and name ca)	30.0	2
assign (segid b and resid 285 and name c) (segid b and resid 286 and name n) 1 -50.0	30.0	2
assign (segid b and resid 286 and name n) (segid b and resid 286 and name ca)	30.0	2
assign (segid b and resid 286 and name c) (segid b and resid 287 and name n) 1 -50.0	30.0	2
assign (segid b and resid 287 and name n) (segid b and resid 287 and name ca)	30.0	2

Program B.I. Program for Calculating CSI

```

/* csi_sv_calc2.c from csi_sv_calc2.c 990122 */
/* Biochemistry 1984, 33, 3287-3295 Y.K.Chøest.al;
J.Mol.Biol. 1986, 264, 1101-1116 R.A.Venters et.al.;
J.A.C.S. 1991, 113, 5490-5492 S.Spera & A.Bax*/

#include<stdio.h>
#include<math.h>
#define fname0 "csi_table"
#define fname1 "csi_std.tbl"

main()
(
int no, noi, i, j, HN2D_id[3000], HN2D_no[3000], CSI_no[3000], csi_ha[3000],
csi_co[3000], csi_ca[3000];
float NOESY_ca[3000], NOESY_cb[3000], HN2D_hn[3000], tocsy_ha[3000], HN2D_co[3000],
delta_cacb[3000], delta_cb[3000], delta_co[3000], delta_ha[3000], csi_ave[3000],
float std_ca[3000], std_cb[3000], std_co[3000], std_ha[3000];
char HN2D_type[3000][3];
FILE *fi;
if ((fi = fopen(fname0, "r")) == NULL)
(
printf("Cannot open file");
)
i = 0;
while (!feof(fi), "%d %s %d %f %f %f %f %f",
&HN2D_id[i], HN2D_type[i], &HN2D_no[i], &NOESY_ca[i], &NOESY_cb[i],
&HN2D_co[i], &tocsy_ha[i], &HN2D_hn[i]) != EOF)
{
noi = i;
fclose(fi);
if ((fi = fopen(fname1, "r")) == NULL)
(
printf("Cannot open file");
)
i = 0;
while (!feof(fi), "%d %s %f %f %f %f",
&std_ha[i], CSI_no[i], CSI_type[i], &std_ca[i], &std_cb[i],
&std_co[i]) != EOF)
{
noi = i;
fclose(fi);
}
/* CSI */
printf("CSI = 0.25*(1*delta_cb[i-1]*2*delta_cb[i]+1*delta_cb[i+1])");
printf("hsqc ID dca dcb dcaCb csi\n\n");
for (i = 0; i < noi-1; i++)
{
for (j = 0; j <= noi-1; j++)
if (strcmp(HN2D_type[i], CSI_type[j]) == 0)
{
delta_ca[i] = NOESY_ca[i] - std_ca[j];
delta_cb[i] = NOESY_cb[i] - std_cb[j];
delta_cacb[i] = delta_ca[i] - delta_cb[j];
csi_ave[i] = 0.25*(1*delta_cacb[i-1] + 2*delta_cacb[i] + 1*delta_cacb[i+1]);
}
}
}

```

Program B.2. Program for Sorting Out Candidates from Assignment Table

```

/* listUP.c revised from listUP.c for envZ1_ASG2.tbl
   sorting out peaks from the assignment table*/

#include<stdio.h>
#include<math.h>
#include<string.h>
#include<stdlib.h>
#define fname0 "envZ1_ASG2.tbl"
#define err_h 0.08
#define err_c 0.50
#define err_n 1.00
#define N_area 100.00

main(argc,argv)
int argc;
float *argv[];
{
    int no, i;
    float cn_shift[1000], h_shift[1000];
    float shift_x, diff_h, diff_cn, cn_comp;
    char res_id[500][5], atom_id[500][5];
    FILE *fi;

    shift_x = atof(argv[1]);

    if ((fi = fopen(fname0, "r")) == NULL)
    {
        printf("Cannot open file");
    }
    i = 1;
    while (fscanf(fi, "%s %s %f %f",
        res_id[i], atom_id[i], &cn_shift[i], &h_shift[i])
        != EOF)
    {
        i++;
        no = i;
        fclose(fi);

        for (i = 1; i <= no; i++)
        {
            diff_h = fabs(shift_x - h_shift[i]);
            if (diff_h <= err_h )
            {
                printf ("%5s %-5s %6.2f %4.2f | dH %6.2f\n",
                    res_id[i], atom_id[i], cn_shift[i], h_shift[i], diff_h);
                diff_cn = fabs(shift_x - cn_shift[i]);
                cn_comp = N_area - shift_x;
                if (cn_comp > 0 && diff_cn <= err_c)
                {
                    printf ("%5s %-5s %6.2f %4.2f | dC %6.2f\n",
                        res_id[i], atom_id[i], cn_shift[i], h_shift[i], diff_cn);
                    if (cn_comp <= 0 && diff_cn <= err_n)
                    {
                        printf ("%5s %-5s %6.2f %4.2f | dN %6.2f\n",
                            res_id[i], atom_id[i], cn_shift[i], h_shift[i], diff_cn);
                    }
                }
            }
        }
    }
}

```

Program B.1. (continued)

```

printf ("%4d %3s %4d %6.2f %6.2f %6.2f %6.2f %6.2f\n", HN2D_id[i], HN2D_type[i],
HN2D_no[i], delta_ca[i], delta_cb[i], delta_cacbi], csi_ave[i]);
}
}

```

Program B.3. Program for Searching Slices

```

7. slice_search_comp.c 990105*
/* this program will show the slice number of peaks for each experiment */

#include<stdio.h>
#include<math.h>
#include<string.h>
#include<stdlib.h>
#define fname0 "slice_c13n.TBL"
#define fname1 "slice_NRESYC.TBL"
#define fname2 "slice_NRESYC2.TBL"
#define fname3 "slice_CN.TBL"
#define fname4 "slice_13NNOE.TBL"
#define fname5 "slice_HOCH.TBL"
#define err_c 0.18 /*err_XXX == 2_diff*/
#define err_nc 0.15
#define err_nc2 0.19
#define err_cnc 0.16
#define err_cnn 0.39
#define err_15N 0.23
#define err_HOCH 0.20

main(argc,argv)
int argc;
float *argv[];
{
    int no, no1, no2, i, slice_no[500], slice_no2[500], slice_no3[500];
    int no3, slice_no4[500], no4, slice_no5[500], no5, slice_no6[500];
    float cshift_1[300], cshift_2[300], cshift_3[300], cshift_4[300];
    float cshift_5[300], shift_x, diff_c1, diff_c2, diff_c3, diff_c4;
    float diff_c5, diff_c6, diff_c7, diff_c8, diff_c9, diff_c10;
    float cshift_6[300], cshift_7[300], cshift_8[300], cshift_9[300];
    float cshift_10[300], cshift_11[300], cshift_12[300], cshift_13[300];
    float cshift_14[300], cshift_15[300];
    float diff_c11, diff_c12, diff_c13, diff_c14, diff_c15;
    float cshift_16[300], cshift_17[300], cshift_18[300], cshift_19[300];
    float cshift_20[300], cshift_21[300];
    float diff_c16, diff_c17, diff_c18, diff_c19, diff_c20, diff_n1;
    float nshift_2[300], nshift_3[300], nshift_4[300], nshift_5[300];
    float nshift_6[300], cshift_21[300], cshift_22[300], cshift_23[300];
    float diff_n2, diff_n3, diff_n4, diff_n5, diff_n6;
    float diff_c21, diff_c22, diff_c23;
    FILE *fi;

    shift_x = atof(argv[1]);
    printf ("GO & SEE \n");
    printf ("-----\n");
    if ((fi = fopen(fname0, "r")) == NULL)
    {
        printf("Cannot open file");
    }
    i = 0;
    while (fscanf(fi, "%d %f %f %f %f",
        &slice_no[i], &cshift_1[i], &cshift_2[i], &cshift_3[i], &cshift_4[i],
        &cshift_5[i]) != EOF)
        i++;
    no = i;

    if ((fi = fopen(fname1, "r")) == NULL)
    {
        printf("Cannot open file");
    }
    while (fscanf(fi, "%d %f %f %f %f",
        &slice_no[i], &cshift_1[i], &cshift_2[i], &cshift_3[i], &cshift_4[i],
        &cshift_5[i]) != EOF)
        i++;
    no = i;

    if ((fi = fopen(fname2, "r")) == NULL)
    {
        printf("Cannot open file");
    }
    while (fscanf(fi, "%d %f %f %f %f",
        &slice_no[i], &cshift_1[i], &cshift_2[i], &cshift_3[i], &cshift_4[i],
        &cshift_5[i]) != EOF)
        i++;
    no = i;

    if ((fi = fopen(fname3, "r")) == NULL)
    {
        printf("Cannot open file");
    }
    while (fscanf(fi, "%d %f %f %f %f",
        &slice_no[i], &cshift_1[i], &cshift_2[i], &cshift_3[i], &cshift_4[i],
        &cshift_5[i]) != EOF)
        i++;
    no = i;

    if ((fi = fopen(fname4, "r")) == NULL)
    {
        printf("Cannot open file");
    }
    while (fscanf(fi, "%d %f %f %f %f",
        &slice_no[i], &cshift_1[i], &cshift_2[i], &cshift_3[i], &cshift_4[i],
        &cshift_5[i]) != EOF)
        i++;
    no = i;

    if ((fi = fopen(fname5, "r")) == NULL)
    {
        printf("Cannot open file");
    }
    while (fscanf(fi, "%d %f %f %f %f",
        &slice_no[i], &cshift_1[i], &cshift_2[i], &cshift_3[i], &cshift_4[i],
        &cshift_5[i]) != EOF)
        i++;
    no = i;
}

```

Program B.3. (continued)

```

fclose(fi);
if ((fi = fopen(fname1, "r")) == NULL)
{
    printf("Cannot open file");
}
i = 0;
while (fscanf(fi, "%d %f %f %f %f",
    &slice_no[i], &cshift_6[i], &cshift_7[i], &cshift_8[i], &cshift_9[i],
    &cshift_10[i]) != EOF)
    i++;
no1 = i;
fclose(fi);
if ((fi = fopen(fname2, "r")) == NULL)
{
    printf("Cannot open file");
}
i = 0;
while (fscanf(fi, "%d %f %f %f %f",
    &slice_no3[i], &cshift_11[i], &cshift_12[i], &cshift_13[i],
    &cshift_14[i], &cshift_15[i]) != EOF)
    i++;
no2 = i;
fclose(fi);
if ((fi = fopen(fname3, "r")) == NULL)
{
    printf("Cannot open file");
}
i = 0;
while (fscanf(fi, "%d %f %f %f %f",
    &slice_no4[i], &cshift_16[i], &cshift_17[i], &cshift_18[i],
    &cshift_19[i], &cshift_20[i], &nshift_1[i]) != EOF)
    i++;
no3 = i;
fclose(fi);
if ((fi = fopen(fname4, "r")) == NULL)
{
    printf("Cannot open file");
}
i = 0;
while (fscanf(fi, "%d %f %f %f %f",
    &slice_no5[i], &nshift_2[i], &nshift_3[i], &nshift_4[i], &nshift_5[i],
    &nshift_6[i]) != EOF)
    i++;
no4 = i;
fclose(fi);
if ((fi = fopen(fname5, "r")) == NULL)
{
    printf("Cannot open file");
}
i = 0;
while (fscanf(fi, "%d %f %f %f %f",
    &slice_no6[i], &cshift_21[i], &cshift_22[i], &cshift_23[i],
    &cshift_24[i]) != EOF)
    i++;
no5 = i;
fclose(fi);
}

```

Program B.3. (continued)

```

for (i = 0; i <= no; i++)
(
    diff_c1 = fabs(shift_x - Cshift_1[i]);
    if (diff_c1 <= err_c)
    (
        printf ("slice %4d of C13noesy (-) \n", slice_no1[i]);
    )
    diff_c2 = fabs(shift_x - Cshift_2[i]);
    if (diff_c2 <= err_c)
    (
        printf ("slice %4d of C13noesy (+) \n", slice_no1[i]);
    )
    diff_c3 = fabs(shift_x - Cshift_3[i]);
    if (diff_c3 <= err_c)
    (
        printf ("slice %4d of C13noesy (-) \n", slice_no1[i]);
    )
    diff_c4 = fabs(shift_x - Cshift_4[i]);
    if (diff_c4 <= err_c)
    (
        printf ("slice %4d of C13noesy (+) \n", slice_no1[i]);
    )
    diff_c5 = fabs(shift_x - Cshift_5[i]);
    if (diff_c5 <= err_c)
    (
        printf ("slice %4d of C13noesy (-) \n", slice_no1[i]);
    )
    diff_c6 = fabs(shift_x - Cshift_6[i]);
    if (diff_c6 <= err_nc)
    (
        printf ("slice %4d of noesy_cpurge(h2o) (-) \n", slice_no2[i]);
    )
    diff_c7 = fabs(shift_x - Cshift_7[i]);
    if (diff_c7 <= err_nc)
    (
        printf ("slice %4d of noesy_cpurge(h2o) (+) \n", slice_no2[i]);
    )
    diff_c8 = fabs(shift_x - Cshift_8[i]);
    if (diff_c8 <= err_nc)
    (
        printf ("slice %4d of noesy_cpurge(h2o) (-) \n", slice_no2[i]);
    )
    diff_c9 = fabs(shift_x - Cshift_9[i]);
    if (diff_c9 <= err_nc)
    (
        printf ("slice %4d of noesy_cpurge(h2o) (+) \n", slice_no2[i]);
    )
    diff_c10 = fabs(shift_x - Cshift_10[i]);
    if (diff_c10 <= err_nc)
    (
        printf ("slice %4d of noesy_cpurge(h2o) (-) \n", slice_no2[i]);
    )
    diff_c11 = fabs(shift_x - Cshift_11[i]);
    if (diff_c11 <= err_nc2)
    (
        printf ("slice %4d of noesy_cpurge(d20) (-) \n", slice_no3[i]);
    )
    diff_c12 = fabs(shift_x - Cshift_12[i]);

```

Program B.3. (continued)

```

if (diff_c12 <= err_nc2)
(
    printf ("slice %4d of noesy_cpurge(d20) (+) \n", slice_no3[i]);
)
diff_c13 = fabs(shift_x - Cshift_13[i]);
if (diff_c13 <= err_nc2)
(
    printf ("slice %4d of noesy_cpurge(d20) (-) \n", slice_no3[i]);
)
diff_c14 = fabs(shift_x - Cshift_14[i]);
if (diff_c14 <= err_nc2)
(
    printf ("slice %4d of noesy_cpurge(d20) (+) \n", slice_no3[i]);
)
diff_c15 = fabs(shift_x - Cshift_15[i]);
if (diff_c15 <= err_nc2)
(
    printf ("slice %4d of noesy_cpurge(d20) (-) \n", slice_no3[i]);
)
diff_c16 = fabs(shift_x - Cshift_16[i]);
if (diff_c16 <= err_cnc)
(
    printf ("slice %4d of CN_noesy (-) \n", slice_no4[i]);
)
diff_c17 = fabs(shift_x - Cshift_17[i]);
if (diff_c17 <= err_cnc)
(
    printf ("slice %4d of CN_noesy (+) \n", slice_no4[i]);
)
diff_c18 = fabs(shift_x - Cshift_18[i]);
if (diff_c18 <= err_cnc)
(
    printf ("slice %4d of CN_noesy (-) \n", slice_no4[i]);
)
diff_c19 = fabs(shift_x - Cshift_19[i]);
if (diff_c19 <= err_cnc)
(
    printf ("slice %4d of CN_noesy (+) \n", slice_no4[i]);
)
diff_c20 = fabs(shift_x - Cshift_20[i]);
if (diff_c20 <= err_cnc)
(
    printf ("slice %4d of CN_noesy (-) \n", slice_no4[i]);
)
diff_n1 = fabs(shift_x - Nshift_1[i]);
if (diff_n1 <= err_cnn)
(
    printf ("slice %4d of CN_noesy (N) \n", slice_no4[i]);
)
diff_n2 = fabs(shift_x - Nshift_2[i]);
if (diff_n2 <= err_15N)
(
    printf ("slice %4d of 15Nnoesy (-) \n", slice_no5[i]);
)
diff_n3 = fabs(shift_x - Nshift_3[i]);
if (diff_n3 <= err_15N)
(
    printf ("slice %4d of 15Nnoesy (+) \n", slice_no5[i]);
)
diff_n4 = fabs(shift_x - Nshift_4[i]);

```

Program B.3. (continued)

```
if (diff_n4 <= err_15N)
{
printf ("slice %4d of 15Nnoesy (-)\n", slice_no5[i]);
diff_n5 = fabs(shift_x - Nshift_5[i]);
if (diff_n5 <= err_15N)
{
printf ("slice %4d of 15Nnoesy (+)\n", slice_no5[i]);
diff_n6 = fabs(shift_x - Nshift_6[i]);
if (diff_n6 <= err_15N)
{
printf ("slice %4d of 15Nnoesy (-)\n", slice_no5[i]);
diff_c21 = fabs(shift_x - Cshift_21[i]);
if (diff_c21 <= err_HCCH)
{
printf ("slice %4d of HCCH-TOCSY (-)\n", slice_no6[i]);
diff_c22 = fabs(shift_x - Cshift_22[i]);
if (diff_c22 <= err_HCCH)
{
printf ("slice %4d of HCCH-TOCSY (+)\n", slice_no6[i]);
diff_c23 = fabs(shift_x - Cshift_23[i]);
if (diff_c23 <= err_HCCH)
{
printf ("slice %4d of HCCH-TOCSY (-)\n", slice_no6[i]);
}
}
}
}
}
}
```

-A79-

Program B.4. Program for Finding Mirror Image

```
/* find_mirror_image2.c
   sorting out the diagonal peaks from the output of capp*/

#include<stdio.h>
#include<math.h>
#include<string.h>
#define fname0 "envZ_HCCH.PCK"
#define tor_x 0.05
#define tor_y 0.05
#define Z_range 23.94 /* sweep width/spectrometer freq.*/

main(argc,argv)
int argc;
char *argv[];
{
int no, i, j, Peak_id[9000];
float Peak_x[9000], Peak_y[9000], Peak_z[9000], Peak_int[9000];
float x_a, y_a, z_a, z_plus, z_minus, int_a, diff_xy, diff_yx;
float z_plus1, z_minus1;
int id_a, slice_a, Slice_no[9000];
FILE *fi;

if(argc != 2)
{
printf("argument 1: id number to be searched\n");
exit(1);
}

id_a = atoi(argv[1]);

if ((fi = fopen(fname0, "r")) == NULL)
{
printf("Cannot open file");
}

i = 1;
while (fscanf(fi, "%d %d %f %f %e",
&Peak_id[i], &slice_no[i], &Peak_x[i], &Peak_y[i], &Peak_z[i],
&Peak_int[i])
!=EOF)
i++;
no = i;
fclose(fi);

for (j = 1; j <= no; j++)
{
if(Peak_id[j] == id_a)
{
x_a = Peak_x[j];
y_a = Peak_y[j];
z_a = Peak_z[j];
z_plus = z_a + Z_range;
z_minus = z_a - Z_range;
slice_a = Slice_no[j];
int_a = Peak_int[j];
}
printf ("%4d %3d %6.2f %6.2f %7.2f (%6.2f/%6.2f) %8.3e\n",
id_a, slice_a, x_a, y_a, z_a, z_plus, z_minus, int_a);
printf ("-----\n");
}
}
```

-A80-

Program B.4. (continued)

```

for (i = 1; i <= no; i++)
{
    diff_xy = fabs(x_a - Peak_y[i]);
    diff_yx = fabs(y_a - Peak_x[i]);
    if (diff_xy <= tor_x && diff_yx <= tor_y)
    {
        z_plus1 = Peak_z[i] + z_range;
        z_minus1 = Peak_z[i] - z_range;
        printf ("%4d %3d %6.2f %6.2f %7.2f (%6.2f/%6.2f) %8.3e\n",
            Peak_id[i], Slice_no[i], Peak_y[i], Peak_x[i], z_plus1,
            z_minus1, Peak_int[i]);
    }
}

```

```

/* H_exchange.c 971125 for H-bond data by hsqc with/without h2o
presat. */
#include<stdio.h>
#define fname0 "hsqc_with_h2o.TBL"
#define fname1 "hsqc_no_h2o.TBL"
#define error_hn 0.0301
#define error_n15 0.101
#define offset_wh_hn 0.000 /* hsqc_hn + offset_hn */
#define offset_wh_n15 0.000 /* hsqc_hx + offset_hx */
#define BigT 0.001375

main()
{
    int no, nol, i, j;
    int hsqc_id[200], hsqc2_id[200], hsqc_slicenum[200];
    int hsqc2_slicenum[200];
    float hsqc_hn_tmp, hsqc_n15_tmp;
    float delta_hn, delta_n15, intrn_ratio, ex_rate;
    float hsqc_hn[200], hsqc2_HN[200], hsqc_n15[200], hsqc2_n15[200], hsqc2_intn[200],
    hsqc2_intn[200];
    FILE *fi;

    if ((fi = fopen(fname0, "r")) == NULL)
    {
        printf("Cannot open file");
    }
    i = 0;
    while (fscanf(fi, "%d %f %f",
        &hsqc_id[i], &hsqc_hn[i], &hsqc_n15[i], &hsqc2_intn[i]) != EOF)
    {
        i++;
        no = i;
        fclose(fi);
    }
    if ((fi = fopen(fname1, "r")) == NULL)
    {
        printf("cannot open file");
    }
    i = 0;
    while (fscanf(fi, "%d %f %f",
        &hsqc2_id[i], &hsqc2_HN[i], &hsqc2_n15[i], &hsqc2_intn[i]) != EOF)
    {
        i++;
        nol = i;
        fclose(fi);
    }
    /* hsqc */
    printf ("hsqc(+h) vs. hsqc(-h) \n\n");
    printf ("OFFSET FOR HN (+h2o presat.): %5.4f \n", offset_wh_hn);
    printf ("OFFSET FOR N (+h2o presat.): %5.4f \n", offset_wh_n15);
    printf ("BigT : %7.6f \n", BigT);
    printf ("\n\n");
    for (i = 0; i <= no-1; i++)
    {
        hsqc_hn_tmp = hsqc_hn[i] + offset_wh_hn;
        hsqc_n15_tmp = hsqc_n15[i] + offset_wh_n15;

```

Program B.5. (continued)

```

for (j = 0; j <= ncl-1; j++)
{
    delta_hn = hsgc2_HN[j] - hsgc_hn_tmp;
    if (delta_hn <= 0) delta_hn = -1*delta_hn;
    delta_n15 = hsgc2_N15[j] - hsgc_n15_tmp;
    if (delta_n15 <= 0) delta_n15 = -1*delta_n15;
    if (delta_hn <= error_hn && delta_n15 <= error_n15)
    {
        intrn_ratio = hsgc_intn[i]/hsgc2_intn[j];
        ex_rate = (1/intrn_ratio - 1)/(BigT*100);
        printf ("%4d %6.2f %6.2f %6.2e\n", hsgc_id[i], hsgc_hn_tmp,
            hsgc_n15_tmp, hsgc_intn[i]);
        printf ("%4d %6.2f %6.2f %6.3f %6.3f\n\n", hsgc2_id[j],
            hsgc2_HN[j], hsgc2_N15[j], hsgc2_intn[j], intrn_ratio, ex_rate);
    }
}
}

```

Program B.6. Program for Analyzing HNHA Data

```

/* hnha_special2.c 976030 */
#include<stdio.h>
#include<math.h>
#define fname0 "envz_HNHA.PCK"
#define fname1 "ASG.tbl"
#define error_hn 0.0301
#define error_hx 0.0301
#define error_n15 0.301
#define error_dhn 0.0501
#define error_dnl5 0.501
#define offset_hn 0.000
#define offset_hx 0.000
#define offset_n15 0.000
#define water_shift 4.77
#define pai 3.14159
#define param 0.0268
#define n15_range 16.44 /* Z_range obtained from Z_SW/Z_SF */

main()
{
    int no, nol, i, j, k, HNHA_id[3000], HNHA_sllicenum[3000], hsgc_id[1000],
    res_id[1000];
    float HNHA_hn_tmp, HNHA_n15_tmp, delta_hn, delta_n15, delta_hx, delta_nx,
    HNHA_hx_tmp;
    float HNHA_hn[3000], HNHA_hx[3000], HNHA_n15[3000], HNHA_intn[3000];
    float HNHA_hn_tmp2, HNHA_hx_tmp2, HNHA_n15_tmp2, delta_hx2, delta_nx2, HNHA_dev_tmp;
    float asg_n15[300], asg_hn[300], asg_co[300], asg_ca[300], asg_ha[300], asg_cb[300],
    asg_hb[300];
    float diff_n15, diff_n15_add, diff_n15_sub, diff_hn;
    char aa_type[300][3];
    double j_hnha;

    FILE *fi;

    if ((fi = fopen(fname0, "r")) == NULL)
    {
        printf("Cannot open file");
    }
    i = 0;
    while (fscanf(fi, "%d %d %: %f %f %f",
        &HNHA_id[i], &HNHA_sllicenum[i], &HNHA_hn[i], &HNHA_hx[i],
        &HNHA_n15[i], &HNHA_intn[i]) != EOF)
        i++;
    no = i;
    fclose(fi);

    if ((fi = fopen(fname1, "r")) == NULL)
    {
        printf("Cannot open file");
    }
    i = 0;
    while (fscanf(fi, "%d %s %d %f %f %f %f %f %f",
        &hsgc_id[i], aa_type[i], &asg_ca[i], &asg_ha[i], &asg_cb[i], &asg_hb[i]) != EOF)
        i++;
    nol = i;
}

```

Program B.6. (continued)

```

fclose(fi);
/* HNHA */
printf ("HNHA list");
printf ("\n\n");
printf (" pk#   sl#   hn   hx   nl5   intrn   - lha/thn");
for (i = 0; i <= no-1; i++)
{
  HNHA_hn_tmp = HNHA_hn[i] + offset_hn;
  HNHA_hx_tmp = HNHA_hx[i] + offset_hx;
  delta_nx = HNHA_hn_tmp - HNHA_hx_tmp;
  if (delta_nx <= 0) delta_nx = -1*delta_nx;
  delta_hx = HNHA_hx[i] - water_shift;
  if (delta_hx <= 0) delta_hx = -1*delta_hx;
  if (delta_hx >= error_hx && delta_nx <= error_hn)
  {
    for (j = 0; j <= no-1; j++)
    {
      if (j != i)
      {
        HNHA_hn_tmp2 = HNHA_hn[j] + offset_hn;
        delta_hn = HNHA_hn_tmp2 - HNHA_hn_tmp;
        if (delta_hn <= 0) delta_hn = -1*delta_hn;
        HNHA_hx_tmp2 = HNHA_hx[j] + offset_hx;
        delta_hx2 = HNHA_hn_tmp2 - HNHA_hx_tmp2;
        HNHA_nl5_tmp = HNHA_nl5[i] + offset_nl5;
        HNHA_nl5_tmp2 = HNHA_nl5[j] + offset_nl5;
        delta_nl5 = HNHA_nl5_tmp2 - HNHA_nl5_tmp;
        if (delta_nl5 <= 0) delta_nl5 = -1*delta_nl5;
        delta_hx2 = HNHA_hx[j] - water_shift;
        if (delta_hx2 <= 0) delta_hx2 = -1*delta_hx2;
        if (delta_hn <= error_hn && delta_nl5 <= error_nl5 && delta_hx2 >= error_hx &&
          delta_hx2 >= error_hn)
        {
          HNHA_dev_tmp = -1*(HNHA_intrn[j]/HNHA_intrn[i]);
          if (HNHA_dev_tmp >= 0)
          {
            j_hnha = (atan(sqrt(HNHA_dev_tmp)))/(pai*param);
            printf ("%4d %4d %6.2f %6.2f %6.2f %6.2e \n", HNHA_id[i],
              HNHA_slicenum[i], HNHA_hn_tmp, HNHA_hx_tmp, HNHA_nl5_tmp, HNHA_intrn[i]);
            printf ("%4d %4d %6.2f %6.2f %6.2f %6.2e %6.3f %6.2f\n", HNHA_id[j],
              HNHA_slicenum[j], HNHA_hn_tmp2, HNHA_hx_tmp2, HNHA_nl5_tmp2, HNHA_intrn[j],
              HNHA_dev_tmp, j_hnha);
          }
          for (k=0; k <= no1-1; k++)
          {
            diff_nl5 = asg_nl5[k] - HNHA_nl5_tmp2;
            if (diff_nl5 < 0) diff_nl5 = -1*diff_nl5;
            diff_nl5_add = asg_nl5[k] - (HNHA_nl5_tmp2 + nl5_range);
            if (diff_nl5_add < 0) diff_nl5_add = -1*diff_nl5_add;
            diff_nl5_sub = asg_nl5[k] - (HNHA_nl5_tmp2 - nl5_range);
            if (diff_nl5_sub < 0) diff_nl5_sub = -1*diff_nl5_sub;
            diff_hn = asg_hn[k] - HNHA_hn_tmp2;
            if (diff_hn < 0) diff_hn = -1*diff_hn;

```

Program B.6. (continued)

```

          if (diff_nl5 <= error_dn15 && diff_hn <= error_dhn)
          {
            printf ( "
          %6.2f \n", res_id[k], aa_type[k], asg_ha[k]);
          if (diff_nl5_add <= error_dn15 && diff_hn <= error_dhn)
          {
            printf ( "
          %6.2f \n", res_id[k], aa_type[k], asg_ha[k]);
          if (diff_nl5_sub <= error_dn15 && diff_hn <= error_dhn)
          {
            printf ( "
          %6.2f \n", res_id[k], aa_type[k], asg_ha[k]);
          }
        }
      }
    }
  }
}
printf ("\n");
}
}
}
}

```

84d %2s

84d %2s

84d %2s

Program B.7. Program for Analyzing HNHB Data

```

1 /* search_hnhb3.c 971208 */
#include<stdio.h>
#define filename "envZ_HNHB.FCK"
#define error_hn 0.0301
#define error_n15 0.0501
#define error_hnhx 0.0301
#define offset_hn 0.000 /* HNHB_hn + offset_hn */
#define offset_n15 0.000 /* HNHB_n15 + offset_n15 */

main()
{
  int nol, i, j, HNHB_id[1000], HNHB_sllicenum[500];
  float delta_hn, delta_hnhx;
  float delta_n15, ratio_hnhx;
  float HNHB_hn[1000], HNHB_hx[1000], HNHB_n15[1000], HNHB_intn[1000];
  FILE *fi;
  if ((fi = fopen(filename, "r")) == NULL)
    printf("Cannot open file");
  i = 0;
  while (fscanf(fi, "%d %d %f %f %f",
                &HNHB_id[i], &HNHB_sllicenum[i], &HNHB_hn[i], &HNHB_hx[i],
                &HNHB_n15[i], &HNHB_intn[i]) != EOF)
    i++;
  nol = i;
  fclose(fi);
  /* HNHB */
  printf ("HNHB \n\n");
  printf ("ERROR FOR HN : %5.4f \n", error_hn);
  printf ("ERROR FOR N15 : %5.4f \n", error_n15);
  printf ("OFFSET FOR HN : %5.4f \n", offset_hn);
  printf ("OFFSET FOR N15: %5.4f \n", offset_n15);
  printf ("\n\n");
  printf (" pk# sl# HN HX n15 intn Ic/Ic");
  for (i = 0; i <= nol-1; i++)
  {
    delta_hnhx = HNHB_hn[i] - HNHB_hx[i];
    if (delta_hnhx <= 0) delta_hnhx = -1*delta_hnhx;
    if (delta_hnhx <= error_hnhx)
    {
      printf ("\n");
      printf ("**%4d %4d %6.2f %6.2f %6.2f %6.2f %6.2e\n", HNHB_id[i],
              HNHB_sllicenum[i], HNHB_hn[i], HNHB_hx[i], HNHB_n15[i], HNHB_intn[i]);
      for (j = 0; j <= nol-1; j++)
      {
        if (j != i)
        {
          delta_n15 = HNHB_n15[i] - HNHB_n15[j];
          if (delta_n15 <= 0) delta_n15 = -1*delta_n15;
          delta_hn = HNHB_hn[i] - HNHB_hn[j];

```

Program B.7. (continued)

```

if (delta_hn <= 0) delta_hn = -1*delta_hn;
if (delta_n15 <= error_n15 && delta_hn <= error_hn )
{
  ratio_hnhx = (HNHB_intn[j]/HNHB_intn[i]);
  printf ("%4d %4d %6.2f %6.2f %6.2f %6.2f %6.2e\n", HNHB_id[j],
          HNHB_sllicenum[j], HNHB_hn[j], HNHB_hx[j], HNHB_n15[j], HNHB_intn[j], ratio_hnhx);
}
}
}
}
}
}

```