

Manual for automated Cortex-Based Alignment in BrainVoyager QX via Autolt 3

Manual for automated Cortex-Based Alignment	1
in BrainVoyager QX via Autolt 3.....	1
<i>Introduction</i>	2
Overview	2
Background.....	2
Versions.....	2
Installation of Autolt.....	3
Installation of script.....	3
<i>Usage</i>	4
Running the cortex-based alignment script.....	4
<i>About</i>	6

Introduction

This is a little manual explaining how to perform cortex-based alignment in BrainVoyager QX in an automated way by using the Autolt script from Draulio de Araujo (Brazil).

Overview

This script performs most of the steps for cortex-based alignment (CBA).

Background

Some of the functionality in BrainVoyager that cannot be scripted in BrainVoyager, can be automated using the keyboard and mouse recording tool 'Autolt'. This is a free tool that can be downloaded from the website (<http://www.autoitscript.com/autoit3/>):

"Autolt v3 is a freeware BASIC-like scripting language designed for automating the Windows GUI. It uses a combination of simulated keystrokes, mouse movement and window/control manipulation in order to automate tasks in a way not possible or reliable with other languages (e.g. VBScript and SendKeys)."

Versions

BrainVoyager QX version: 1.10
Autolt version: 3
Platform: Windows XP (not all computers)
Manual: v0.1

Installation of Autolt

Follow the instructions.

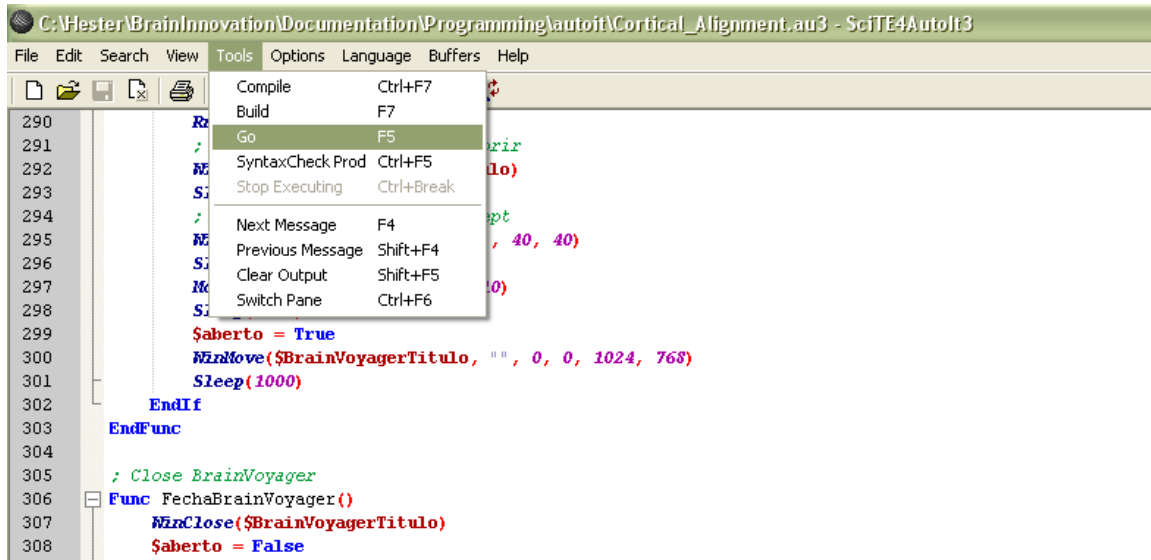
Installation of script

The script does not need to be installed. When Autolt is installed, it will run when double-clicking on the script.

Usage

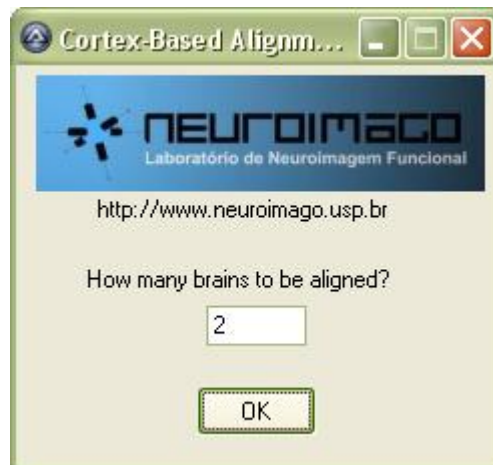
Running the cortex-based alignment script

Once you run it, and select the needed files you, don't touch the mouse anymore. It will do the job. It is tested under different resolutions, and there are still problems with widescreen laptops. It is working fine for the English and Portuguese Windows versions. Other languages might have problems due to the path where BV_.exe is located.

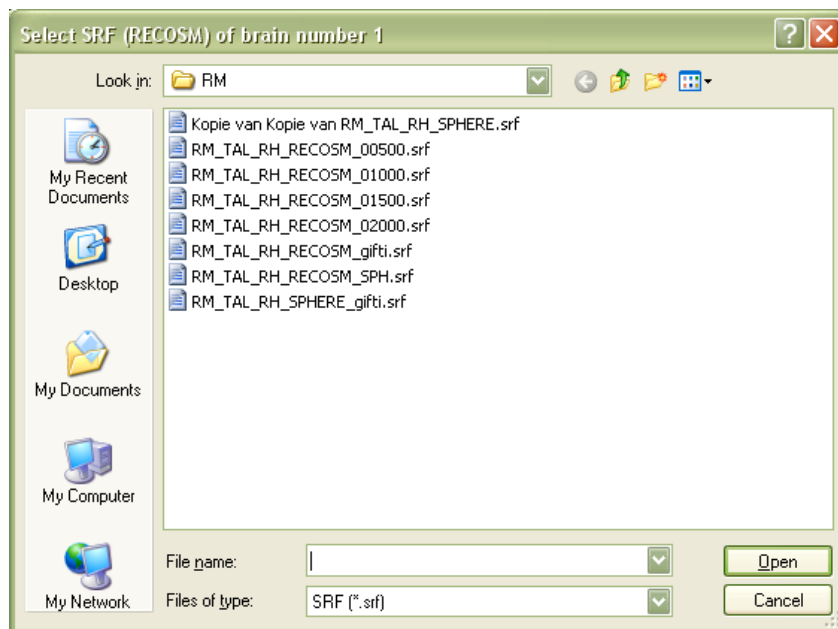


```
290 ;
291 ;
292 ;
293 ;
294 ;
295 ;
296 ;
297 ;
298 ;
299 $aberto = True
300 Move($BrainVoyagerTitulo, "", 0, 0, 1024, 768)
301 Sleep(1000)
302 EndIf
303 EndFunc
304
305 ; Close BrainVoyager
306 Func FechaBrainVoyager()
307     WinClose($BrainVoyagerTitulo)
308     $aberto = False
309     ;
310     ;
311     ;
312     ;
313     ;
314     ;
315     ;
316     ;
317     ;
318     ;
319     ;
320     ;
321     ;
322     ;
323     ;
324     ;
325     ;
326     ;
327     ;
328     ;
329     ;
330     ;
331     ;
332     ;
333     ;
334     ;
335     ;
336     ;
337     ;
338     ;
339     ;
340     ;
341     ;
342     ;
343     ;
344     ;
345     ;
346     ;
347     ;
348     ;
349     ;
350     ;
351     ;
352     ;
353     ;
354     ;
355     ;
356     ;
357     ;
358     ;
359     ;
360     ;
361     ;
362     ;
363     ;
364     ;
365     ;
366     ;
367     ;
368     ;
369     ;
370     ;
371     ;
372     ;
373     ;
374     ;
375     ;
376     ;
377     ;
378     ;
379     ;
380     ;
381     ;
382     ;
383     ;
384     ;
385     ;
386     ;
387     ;
388     ;
389     ;
390     ;
391     ;
392     ;
393     ;
394     ;
395     ;
396     ;
397     ;
398     ;
399     ;
400     ;
401     ;
402     ;
403     ;
404     ;
405     ;
406     ;
407     ;
408     ;
409     ;
410     ;
411     ;
412     ;
413     ;
414     ;
415     ;
416     ;
417     ;
418     ;
419     ;
420     ;
421     ;
422     ;
423     ;
424     ;
425     ;
426     ;
427     ;
428     ;
429     ;
430     ;
431     ;
432     ;
433     ;
434     ;
435     ;
436     ;
437     ;
438     ;
439     ;
440     ;
441     ;
442     ;
443     ;
444     ;
445     ;
446     ;
447     ;
448     ;
449     ;
450     ;
451     ;
452     ;
453     ;
454     ;
455     ;
456     ;
457     ;
458     ;
459     ;
460     ;
461     ;
462     ;
463     ;
464     ;
465     ;
466     ;
467     ;
468     ;
469     ;
470     ;
471     ;
472     ;
473     ;
474     ;
475     ;
476     ;
477     ;
478     ;
479     ;
480     ;
481     ;
482     ;
483     ;
484     ;
485     ;
486     ;
487     ;
488     ;
489     ;
490     ;
491     ;
492     ;
493     ;
494     ;
495     ;
496     ;
497     ;
498     ;
499     ;
500     ;
501     ;
502     ;
503     ;
504     ;
505     ;
506     ;
507     ;
508     ;
509     ;
510     ;
511     ;
512     ;
513     ;
514     ;
515     ;
516     ;
517     ;
518     ;
519     ;
520     ;
521     ;
522     ;
523     ;
524     ;
525     ;
526     ;
527     ;
528     ;
529     ;
530     ;
531     ;
532     ;
533     ;
534     ;
535     ;
536     ;
537     ;
538     ;
539     ;
540     ;
541     ;
542     ;
543     ;
544     ;
545     ;
546     ;
547     ;
548     ;
549     ;
550     ;
551     ;
552     ;
553     ;
554     ;
555     ;
556     ;
557     ;
558     ;
559     ;
560     ;
561     ;
562     ;
563     ;
564     ;
565     ;
566     ;
567     ;
568     ;
569     ;
570     ;
571     ;
572     ;
573     ;
574     ;
575     ;
576     ;
577     ;
578     ;
579     ;
580     ;
581     ;
582     ;
583     ;
584     ;
585     ;
586     ;
587     ;
588     ;
589     ;
590     ;
591     ;
592     ;
593     ;
594     ;
595     ;
596     ;
597     ;
598     ;
599     ;
600     ;
601     ;
602     ;
603     ;
604     ;
605     ;
606     ;
607     ;
608     ;
609     ;
610     ;
611     ;
612     ;
613     ;
614     ;
615     ;
616     ;
617     ;
618     ;
619     ;
620     ;
621     ;
622     ;
623     ;
624     ;
625     ;
626     ;
627     ;
628     ;
629     ;
630     ;
631     ;
632     ;
633     ;
634     ;
635     ;
636     ;
637     ;
638     ;
639     ;
640     ;
641     ;
642     ;
643     ;
644     ;
645     ;
646     ;
647     ;
648     ;
649     ;
650     ;
651     ;
652     ;
653     ;
654     ;
655     ;
656     ;
657     ;
658     ;
659     ;
660     ;
661     ;
662     ;
663     ;
664     ;
665     ;
666     ;
667     ;
668     ;
669     ;
670     ;
671     ;
672     ;
673     ;
674     ;
675     ;
676     ;
677     ;
678     ;
679     ;
680     ;
681     ;
682     ;
683     ;
684     ;
685     ;
686     ;
687     ;
688     ;
689     ;
690     ;
691     ;
692     ;
693     ;
694     ;
695     ;
696     ;
697     ;
698     ;
699     ;
700     ;
701     ;
702     ;
703     ;
704     ;
705     ;
706     ;
707     ;
708     ;
709     ;
710     ;
711     ;
712     ;
713     ;
714     ;
715     ;
716     ;
717     ;
718     ;
719     ;
720     ;
721     ;
722     ;
723     ;
724     ;
725     ;
726     ;
727     ;
728     ;
729     ;
730     ;
731     ;
732     ;
733     ;
734     ;
735     ;
736     ;
737     ;
738     ;
739     ;
740     ;
741     ;
742     ;
743     ;
744     ;
745     ;
746     ;
747     ;
748     ;
749     ;
750     ;
751     ;
752     ;
753     ;
754     ;
755     ;
756     ;
757     ;
758     ;
759     ;
760     ;
761     ;
762     ;
763     ;
764     ;
765     ;
766     ;
767     ;
768     ;
769     ;
770     ;
771     ;
772     ;
773     ;
774     ;
775     ;
776     ;
777     ;
778     ;
779     ;
780     ;
781     ;
782     ;
783     ;
784     ;
785     ;
786     ;
787     ;
788     ;
789     ;
790     ;
791     ;
792     ;
793     ;
794     ;
795     ;
796     ;
797     ;
798     ;
799     ;
800     ;
801     ;
802     ;
803     ;
804     ;
805     ;
806     ;
807     ;
808     ;
809     ;
810     ;
811     ;
812     ;
813     ;
814     ;
815     ;
816     ;
817     ;
818     ;
819     ;
820     ;
821     ;
822     ;
823     ;
824     ;
825     ;
826     ;
827     ;
828     ;
829     ;
830     ;
831     ;
832     ;
833     ;
834     ;
835     ;
836     ;
837     ;
838     ;
839     ;
840     ;
841     ;
842     ;
843     ;
844     ;
845     ;
846     ;
847     ;
848     ;
849     ;
850     ;
851     ;
852     ;
853     ;
854     ;
855     ;
856     ;
857     ;
858     ;
859     ;
860     ;
861     ;
862     ;
863     ;
864     ;
865     ;
866     ;
867     ;
868     ;
869     ;
870     ;
871     ;
872     ;
873     ;
874     ;
875     ;
876     ;
877     ;
878     ;
879     ;
880     ;
881     ;
882     ;
883     ;
884     ;
885     ;
886     ;
887     ;
888     ;
889     ;
890     ;
891     ;
892     ;
893     ;
894     ;
895     ;
896     ;
897     ;
898     ;
899     ;
900     ;
901     ;
902     ;
903     ;
904     ;
905     ;
906     ;
907     ;
908     ;
909     ;
910     ;
911     ;
912     ;
913     ;
914     ;
915     ;
916     ;
917     ;
918     ;
919     ;
920     ;
921     ;
922     ;
923     ;
924     ;
925     ;
926     ;
927     ;
928     ;
929     ;
930     ;
931     ;
932     ;
933     ;
934     ;
935     ;
936     ;
937     ;
938     ;
939     ;
940     ;
941     ;
942     ;
943     ;
944     ;
945     ;
946     ;
947     ;
948     ;
949     ;
950     ;
951     ;
952     ;
953     ;
954     ;
955     ;
956     ;
957     ;
958     ;
959     ;
960     ;
961     ;
962     ;
963     ;
964     ;
965     ;
966     ;
967     ;
968     ;
969     ;
970     ;
971     ;
972     ;
973     ;
974     ;
975     ;
976     ;
977     ;
978     ;
979     ;
980     ;
981     ;
982     ;
983     ;
984     ;
985     ;
986     ;
987     ;
988     ;
989     ;
990     ;
991     ;
992     ;
993     ;
994     ;
995     ;
996     ;
997     ;
998     ;
999     ;
1000    ;
```

The script will ask for the number of brains to be aligned (see figure below).



The script will ask for anatomical (*.vmr) and folded surface (*.RECOSM.srf) files for as many files as brains that need to be aligned.



About...

By Draulio de Araujo
Laboratorio de Neuroimagem Funcional, Brazil.

