UPPSALA
UNIVERSITET

MSc BIOINF 12 009

Adaptive evolution in birds

Nasir Mahmood Abbasi

Degree project in bioinformatics, 2012
Examensarbete i bioinformatik 45 hp till masterexamen, 2012
Biology Education Centre, Uppsala University, and Stockholm Bioinformatics Centre, SciLifeLab, Stockholm
Supervisors: Erik Sonnhammer and Kristoffer Forslund

Abstract

Two different complete bird genomes are available, chicken (*G. gallus)* and zebrafinch (*T. guttata).* They are important for this study because they may contain examples of functions that are adaptively evolved. We want to find evidence on how changes in chicken and zebrafinch lifestyles may have affected their genomes.

Genes are orthologous if their histories separated after a speciation event and if their histories separated after the occurrence of duplication then they are said to be paralogous genes. Genes that are duplicated after a speciation event are inparalogs to each other.

In this study, we wanted to test the hypothesis that gene duplication makes adaptive evolution more likely, i.e. genes in clusters with many inparalogs will be more likely to have experienced positive selection than the genes which have not undergone duplication, i.e. 1-1 clusters. Further, we also looked for gene categories which were enriched in adaptively evolved clusters.

To test the hypothesis we acquired sequences for chicken and zebrafinch from a publically available database, using BioMart. Inparanoid algorithm was used to build inparanoid clusters, which were aligned. Phylogenetic trees were then built for each aligned gene cluster. Furthermore, each phylogenetic tree was tested for consistency in matching with its inparanoid cluster. Once we got alignment files and consistent tree files for each cluster, we used codeml in PAML (Phylogenetic Analysis by Maximum Likelihood) to test for possible adaptive evolution among consistent gene clusters. We performed Likelihood ratio tests on the likelihood values from the codeml using a script to get p values for all those adaptively evolved clusters. Benjamini-Hochberg False Discovery correction tests were applied on adaptively evolved clusters from codeml and clusters with p values less than 0.05 were accepted as adaptively evolved clusters, which include both duplicated and non-duplicated clusters.

Once we got adaptively evolved clusters for this analysis, we performed Fisher's Exact Test for proportions to check if the proportion of clusters containing duplicated genes showing significant adaptive evolution is different from the corresponding proportion among clusters containing non-duplicated genes. From this test we found out that clusters containing duplicated genes which are adaptively evolved have higher likelihood to be adaptively evolved than non-duplicated gene clusters. So, we can conclude that duplications make adaptive evolution more likely, in accordance with our initial hypothesis.

We also tried to investigate whether some gene categories are adaptively evolved more often over others. For this pupose, we did comparisons and found GO terms which are overrepresented in case of adaptively evolved duplicated gene clusters. Our analyses of adaptively evolved duplicated gene clusters revieled the same GO terms that previous analysis had found, such as cell adhesion, cytoskeleton, calcium ion binding and terms related to the extracellular matrix. In both chicken and zebrafinch, we found similar biological processes and cellular components terms enriched, while the exact molecular function can still not be concluded due to the fact that we have found enriched terms with few number of genes and $p>10^{-5}$.

Adaptive evolution in birds

Popular science summary

Nasir Mahmood Abbasi

There are about 10,000 species of birds. Birds genomes are compact containing less DNA and fewer repeats than mammals. After the complete sequencing of chicken (*G. gallus)* and zebrafinch (*T. guttata)*, researchers are now looking ahead to try to find out how the different lifestyles of these species have affected their genomes.

This study was conducted to test the hypothesis that duplication makes adaptive evolution more likely. Further, we wanted to look for what gene categories would possibly be enriched, in case of adaptively evolved gene clusters in these bird species.

To test the hypothesis, species sequences were acquired from a publically available database. Genes from closely related bird species were clustered together using an algorithm and an outgroup was specified to root trees and to strengthen the phylogenetic analysis. Species gene sequences were aligned and two different tree building tools were applied to the alignments to make phylogenetic trees. Phylogenetic trees along with aligned sequences are important for finding potential adaptively evolved gene clusters by comparing different evolutionary models. A statistical test using R programming language was applied on adaptively evolved gene clusters to find out if the proportion of clusters containing duplicated genes showing adaptive evolution differed from the corresponding proportion among clusters containing non-duplicated gene clusters. We have also looked for genes functional categories which were enriched in case of adaptively evolved duplicated gene clusters.

We found that duplicated clusters have higher chance for adaptive evolution and thus we have proven our hypothesis, i.e. duplications make adaptive evolution more likely.

We also tried to investigate if some functional categories are adaptively evolved in our comparisons more often than others. We got GO terms which are overrepresented in case of adaptively evolved duplicated gene clusters. We also found GO terms which were similar to terms which are found by earlier groups. In both chicken and zebrafinch, we found similar prevalent biological process and cellular components terms enriched while the exact molecular function can still not be concluded due to the fact that we have found enriched terms with few number of genes and $p>10^{-5}$.

Degree Project, Master Program in Bioinformatics (45 hp), Autumn 2012, Uppsala University
Stockholm Bioinformatics Centre(SBC), SciLifeLab, Stockholm
Supervisor: Erik Sonnhammer and Kristoffer Forslund

Table of contents

List of tables

List of figures

1 Introduction

Fitness is the ability of individual, as a consequence of their phenotype encoded by different genes, to pass on their alleles to the next generation. If a mutant in the population has a higher relative fitness, it is likely that the allele frequency of that mutant increases and eventually becomes fixed in the population. Such a process, in which alleles important for survival and fitness increase in frequency, is called adaptive evolution or positive selection. Adaptive evolution may help us in finding what genes, parts of gene or gene regions actually matter for a given function [1].

Homology is important for the study of adaptive evolution. We can search for evidence for adaptive evolution by comparing homologous sequences with each other. All the genes which evolved from a common ancestor, irrespective of whether the genes exist in one given species or in different species are called homologs. Genes are paralogous if their histories separated after the occurrence of a duplication of the gene and they are orthologous if their histories separated due to speciation event.

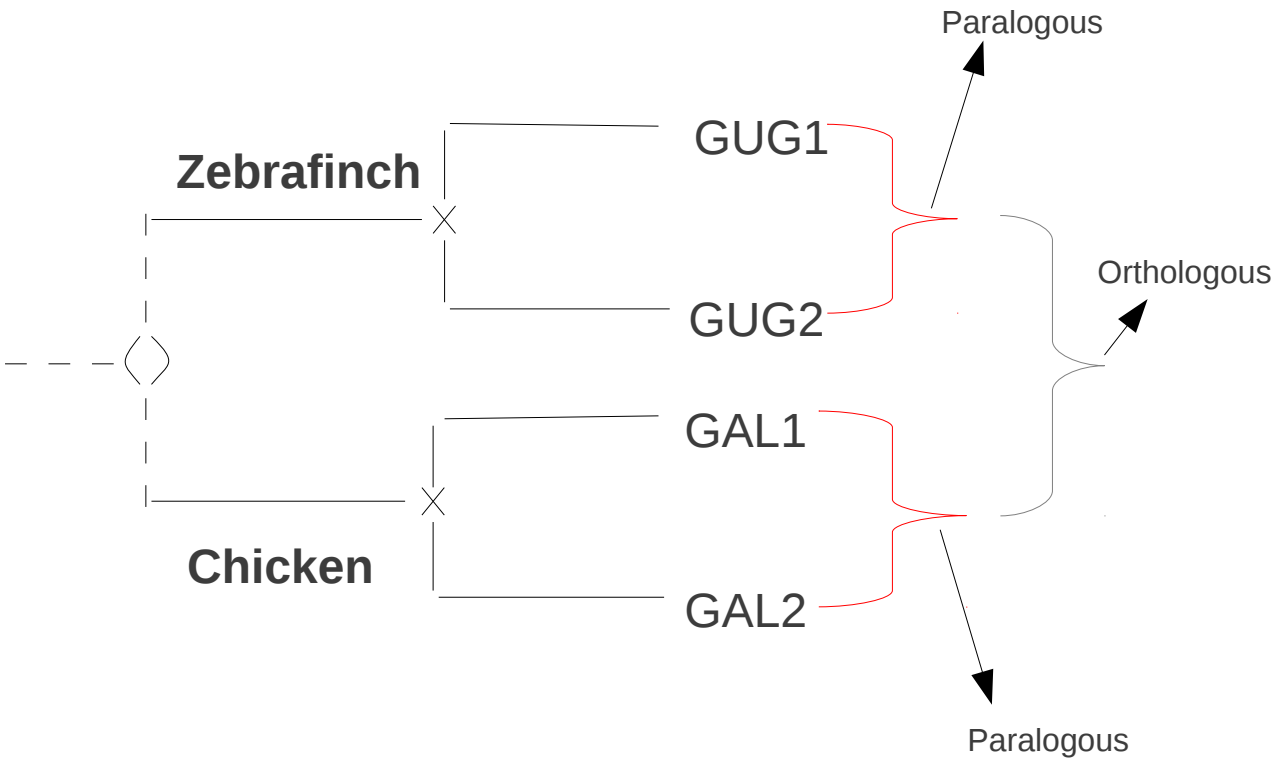

Figure 1: The difference between orthologous and paralogous genes

In the figure 1 GUG represent zebrafinch and GAL represent chicken, Red line shows that zebrafinch and chicken duplicated genes are paralogous while black line shows chicken genes are orthologous to zebrafinch genes. Zebrafinch genes are orthologous to chicken genes because GUG1 and GUG2 are separated after the speciation event while GUG1 is inparalogous to GUG2 and GAL1 is inparalogous to GAL2. These genes are called inparalogs because their histories are

separated after the speciation event and if their histories would be separated before the speciation event then we call them outparalogs.

Duplication can lead to different fates such as pseudogenization, subfunctionalization and neofunctionalization [5]. Pseudogenization is a process in which a functional gene becomes a pseudogene and the gene thus looses its function after sometime. This is common occurrence unless the new gene copy under selection after duplication of the gene [5]. Subfunctionalization is a process in which two copies of the duplicated gene subspecialize to become differentially efficient with regard to two different functions which were already performed by ancestral gene [5]. Neofunctionalization is a process in which there would be emergence of related function or sometimes a completely novel function from genes which are duplicated. We are mainly interested in examples of neofunctionalization to test our hypothesis.

Studying adaptive evolution in birds is interesting because we know according to previous studies that bird genomes are compact, containing less DNA and less repeats than mammals [2]. We do see differences in the genomes of the birds due to their lifestyle which is described in detail in the coming text.

1.1 Zebrafinch genome

Zebrafinches (*T. guttata*) are examples of song birds (though there are many other song birds as well), since they have the ability to communicate through learned vocalizations [3]. Zebrafinches learn these from their fathers during their childhood. Singing in zebrafinch is under control of the neural circuits of the forebrain. This singing behaviour is more prevalent in males than in females [3]. Different genes and gene regions are also involved in gene regulatory processes which might be functionally important for the study. Another important aspect is that learned vocal communications which we see in song birds is important for their reproductive success and it has evolved after divergence of the song bird lineage from other lineages. In zebra finch the genes which are involved in this process seem to be under positive selection [3].

1.2 Chicken genome

Chicken (*G. gallus)* was the first bird whose whole genome was sequenced. The chicken genome is important for studies because we see many changes in the chicken genome due to their lifestyle. From ealier studies, we know that enriched genes and gene families in chicken seem to play roles in immunity and host defence among other things [4].

1.3 Purpose of studying adaptive evolution in birds

The purpose of studying adaptive evolution in birds in our hands was to test our hypothesis that gene duplications will make adaptive evolution more likely, i.e genes in clusters with many inparalogs will be more likely to have experienced positive selection than genes which have not undergone duplication, i.e. 1-1 clusters. This is consistent with the idea that orthologs which come after speciation retain function better than paralogs in which we might see change in the function due to duplication of genes. If that is true, then orthology clusters with more duplications would be enriched for positive selection when compared to clusters with fewer duplications. We wanted to test whether orthologs better retain the conserved functions, and whether adaptive evolution is connected with gene duplication. If this is true, we can say that it is less likely that adaptive evolution occur until a gene duplication event has taken place.

2 Materials and methods

All of the computational work for this study was done locally using the Stockholm Bioinformatics Centre (SBC) and Center of Parallel Computing (PDC) resources. To find adaptively evolved clusters, phylogenetic analysis was performed to study evolutionary history and the changes which occurs in the protein and their functions. To test out hypothesis we performed number of steps. The proposed workflow for these steps is shown in figure 4.

2.1 Data collection

It is difficult to move data from one place to another when you want to query a database. There should be an advanced query interface by which we can easily query data from the database.

2.1.1 BioMart

BioMart [11] was one of the tools which helped us to solve this problem. It is an open source data management system that can be used to group data and to refine data based on different criteria chosen by the user [11]. It has a user friendly web interface which interact with different software packages and allows biologists who don't know programming to use BioMart to query these databases [11]. Chicken, zebrafinch and a suitable outgroup data was collected from Ensembl using BioMart.
Perl script was applied to produce clean versions of those files, by removing genes without chromosome assignments and to check whether nucleotide sequences actually match with the protein sequences under the genetic code.

2.2 Orthologs identification

Phylogenetic analysis of molecular data assumes that the proteins which are under study are indeed homologous. Every analysis assumes that the proteins studied are related by descent to the same ancestral protein. InParanoid 7 [14] was used for identification of orthologs.

2.2.1 InParanoid approach

InParanoid 7 [12] was used to make ortholog groups by clustering pairwise relationships between genes [12]. To get orthologous clusters for both chicken and zebrafinch, a suitable representative protein was selected for each gene in both species. In many such cases each gene in practice has several splice forms. InParanoid approach was used in each case to pick the longest splice form. For each splice form, nucleotides and the protein sequences encoded by them were selected and mitochondrial genes and genes with undefined chromosomal positions were pruned away.

2.3 Outgroup specification

An unrooted phylogenetic tree is a tree without any determined direction of evolution. Correct phylogeny cannot be predicted using unrooted tree. Outgroup specification is important for prediction of correct phylogeny. An outgroup is an external point of reference that should be as closely related to the ingroups as possible without being a member of this ingroup. To find a suitable outgroup for each cluster which contained chicken and zebrafinch genes, all cluster members were blasted against the human proteome and the average bit score was taken for all these comparisons. The human sequence with the highest average bit score to the cluster members was then used as outgroup for every phylogenetic analysis.

2.4 Sequence alignments

Sequence alignment of clusters is an important step for getting good phylogenetic trees. Fasta sequence for genes and proteins were collected for further analysis. Before applying a sequence alignment using Kalign [15], arrangement of ingroup species and outgroup species in an order containing chicken, zebrafinch and human fasta sequences of each cluster in a single text file was done using a custom Perl script. Two types of cluster files exist, duplicated cluster files contain either two or more chicken or zebrafinch gene sequences and non-duplicated have one gene fasta sequence for each of those species. Each gene sequences containing cluster file was indexed with a cluster number. This step would be done for both nucleotide and protein sequences. Now we got 10796 cluster files containing nucleotide and protein gene sequences named with their cluster numbers.

Next step was to do the alignment of protein fasta files using Kalign [15]. It was very hard to do alignments manually for such a large data so bash script was made which used Kalign to do multiple sequence alignment of protein fasta files for all clusters. The bash script contain a following line with a default parameters.

Kalign -i $f -o $f

-i is for input file,-o is for output file

Aligned protein fasta sequence files were then used for alignment of nucleotide fasta sequences files based on protein alignment using a custom perl script. A custom perl script aligned the corresponding codons according to how the amino acid sequences are aligned and included gaps in the positions where protein fasta sequence have them after alignment.

2.5 Alignment trimming

Multiple sequence alignments for many proteins and polynucleotides contain regions which are poorly aligned. Removal of such poorly aligned regions is very important for phylogenetic analyses. So, Alignment trimming is done by using G-Blocks [16]. It removes regions which are considered to be poorly aligned and are not homologous.

2.6 Construction of phylogenetic trees

Trees are the most commonly used representation of evolutionary relationships. To construct the phylogenetic trees, two methods were used. RAxML (Randomized Axelerated Maximum Likelihood Method) [6] which is a maximum likelihood method and FastTree 2 [7] which is minimum evolution method. RAxML has a problem that it could not work well for constructing trees where there are less number of taxa so we used FastTree 2. Duplicated cluster trees were made by RaxML while non-duplicated were made by FastTree 2 which were then used for finding adaptively evolved gene clusters.

Tree construction with RAxML requires change in alignment format because RAxML accepts both protein and nucleotide sequence alignment in PHYLIP format. Different options used RAxML are

RAXMLHPC -s protein/nucleotide.phy -n A1 -m PROTGAMMAWAG

The option -s specifies the sequence file in PHYLIP format which can either be protein or nucleotide. The option -n specifies the suffix which will be added to the end of all the output files. The option -m specifies the model of sequence evolution. PROT in -m part was used for protein, GAMMA for accounting rate heterogeneity among sites and WAG is used for amino acid

substitution matrix. The option -m would be changed according to which data we are using either protein or nucleotide sequence [18].

Bash script was made to run RAxML on aligned duplicated gene clusters to get best phylogenetic trees for duplicated gene clusters. Another bash script was made to run FastTree 2 with its default parameters on the non-duplicated gene clusters. After tree building using these methods, the next step was to find out positively selected clusters using the Codeml program in PAML.

2.7 Consistency checking and bootstrapping

Cluster trees must have chicken, zebrafinch and human as an outgroup in our case in correct order as it is in inparanoid cluster. Figure 2 shows a consistent tree where the trees match with implied topology of inparanoid clusters, while figure 3 shows an inconsistent tree where the cluster tree does not match with implied topology of inparanoid topology.

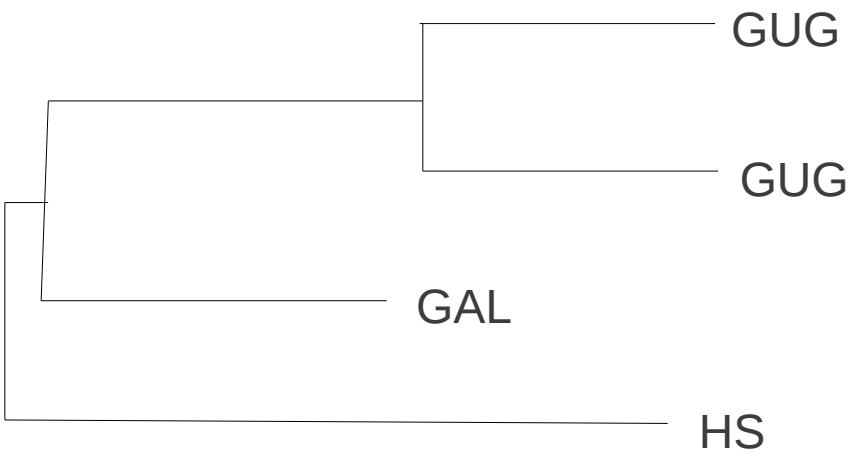

Figure 2: Consistent clusters

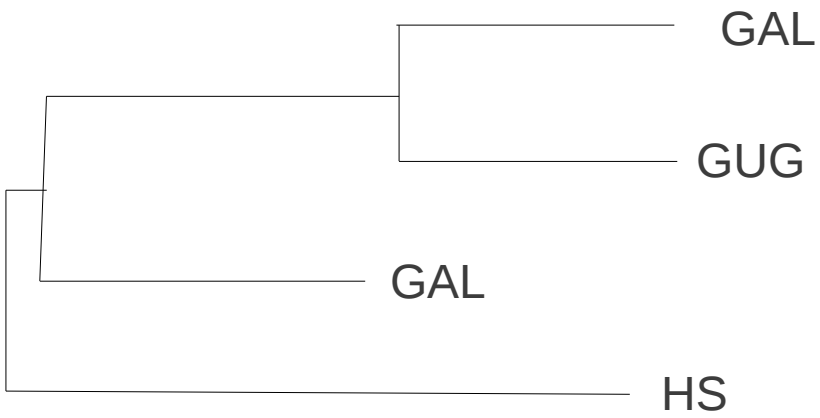

Figure 3: Inconsistent clusters

A custom made perl script was applied on all those cluster trees to check if the cluster trees match the implied topology of the inparanoid clusters, Phylogenetic trees more found to be inconsistent with this script were further tested by bootstrapping to check whether they are actually inconsistent or there is an error in alignment or inparanoid clusters. RAxML was used for bootstrapping by using option -b which is used for random number generation and -# for number of replicates to run [18]. Hundred replicates were run because of few species in our analysis. Inconsistent clusters with very low bootstrap support were discarded and consistent clusters were retained for further analysis. All non duplicated clusters were found to be consistent by use of the Perl script. Statistics of the clusters after consistency checking using custom Perl script and bootstrapping in RAxML, are shown in Table 1.

Table 1: Different statistics of clusters after consistency checking

Total clusters	Clusters with no outgroup	Duplicated clusters	Non duplicated clusters	Total trees	Consistent duplicated clusters	Inconsistent duplicated clusters
10961	165	99	10697	10796	69	30

2.8 Prediction of adaptively evolved gene clusters

PAML, Phylogenetic Analysis by Maximum Likelihood, is a programming package that contain numerous sub-programs [8][9]. The program which is important for finding adaptively evolved clusters is Codeml. This program reads its execution parameters from a control file.

Codeml control files must list 1) alignment file in PHYLIP format, 2) a tree file which should be in newick format. Control file was made for each cluster and these options were changed for each control file using custom perl script. Outfile, Seqtype, Codonfreq, model and NSsites were changed once in the initial file and then custom Perl script is applied which will make a separate control file for each analysis. Other options in control file were used as default.

To predict adaptively evolved clusters in the chicken and zebrafinch, a custom perl script is used which has made a folder for each of those clusters and copy the sequence file, tree file and the control files once it is changed into that folder. The folder was named with the cluster name so it would be easy to analyze result. Codeml program in PAML was run using custom bash script on sequence file,tree file and changed control files and we will get number of output result files.

The output from codeml includes a number of output result files but we were mainly interested in the output file which we named in the control file. A custom Perl script was made to take a log likelihood value from that file for each model and then apply a likelihood ratio test using this formula:

$$LRT = |(2 * (-a - -b))|$$

Perl script takes the likelihood ratio test value for each model and degree of freedom by looking at the number of parameters used by both models, P value was calculated by applying a χ^2 test in R for all clusters. To find the adaptively evolved clusters, Benjamini-Hochberg FDR (false discovery rate) correction test [13] was applied. Benjamini-Hochberg FDR correction test was applied using a custom Perl script which has retained all those clusters which have values less than 0.05. Statistics

of adaptively evolved clusters are shown in the Table 2.

Table 2: Statistics of adaptively evolved clusters

Total no of clusters	Total no of trees	Adaptively evolved duplicated clusters	Adaptively evolved non-duplicated clusters
10766	10766	14	891

2.9 FDR analysis

FDR (false discovery rate) correction using the Benjamini-Hochberg method [13] was applied when multiple tests were performed because when we were doing multiple tests for a large number of hypotheses. There is a risk that we commit type I error, i.e. we falsely predict clusters which should not be selected. To apply FDR (false discovery rate) correction using the Benjamini-Hochberg method [13] a text file was made in which one column was the cluster names and second with their P values. A custom made Perl script run the the Benjamini-Hochberg FDR correction test on that file which have given us true clusters having values less than 0.05.

Gene Ontology (GO) [10] provides a structural vocabulary for annotation of genes and proteins. GO terms are structured in a hierarchy, ranging from more general to more specific. GO is structured in three ontologies,biochemical function, cellular component and molecular function [14].

For the purpose of GO term enrichment analysis three comparisons were made which are further explained in Table 3.

2.10 Gene ontology term enrichment analysis

Table 3: Comparisons to find enriched GO terms

Comparison 1	Adaptive (Including both duplicated and non-duplicated genes which are positively selected)	Nonadaptive (All other genes which are not positively selected)
Comparison 2	Duplicated genes (without considering adaptation)	non-duplicated genes (all three taxa clusters)
Comparison 3	Adaptively duplicated genes	Nonadaptive and nonduplicative genes

In all these comparisons, two separate files were made using a perl script for all genes in both groups. This Perl script matches these genes with gene ontology files from Ensembl database and put both genes and their GO terms in one file. Similarly, same procedure was applied to reference group. Now, when we got genes for both groups along with their GO terms, excel was used to find the unique GO terms which were present in both groups to find the enriched term in either group. After getting the unique GO terms, custom Perl script was made which have counted the number of occurrences of each GO term in each of those groups and save them in separate files. For each GO term, we tested whether the two groups differ in the frequency of that GO term using Fisher's exact test. Fisher's exact test returns the p-value. However, we cannot directly use the individual p-value for each GO term, because we were testing multiple hypotheses, one for each term.

There are several methods available to account for multiple testing. We have selected Benjamini-Hochberg FDR correction test [13] to control the False Discovery Rate (FDR). To apply a Benjamini-Hochberg FDR correction test we have made a text file which has first column of GO terms and second column of p-value which we got from Fisher's exact test. Benjamini-Hochberg FDR correction test for $P<0.05$ was applied on that text file to find out which GO terms are enriched in adaptive (including both duplicated and non-duplicated genes), duplicated genes (without considering adaptation) and adaptively duplicated genes.

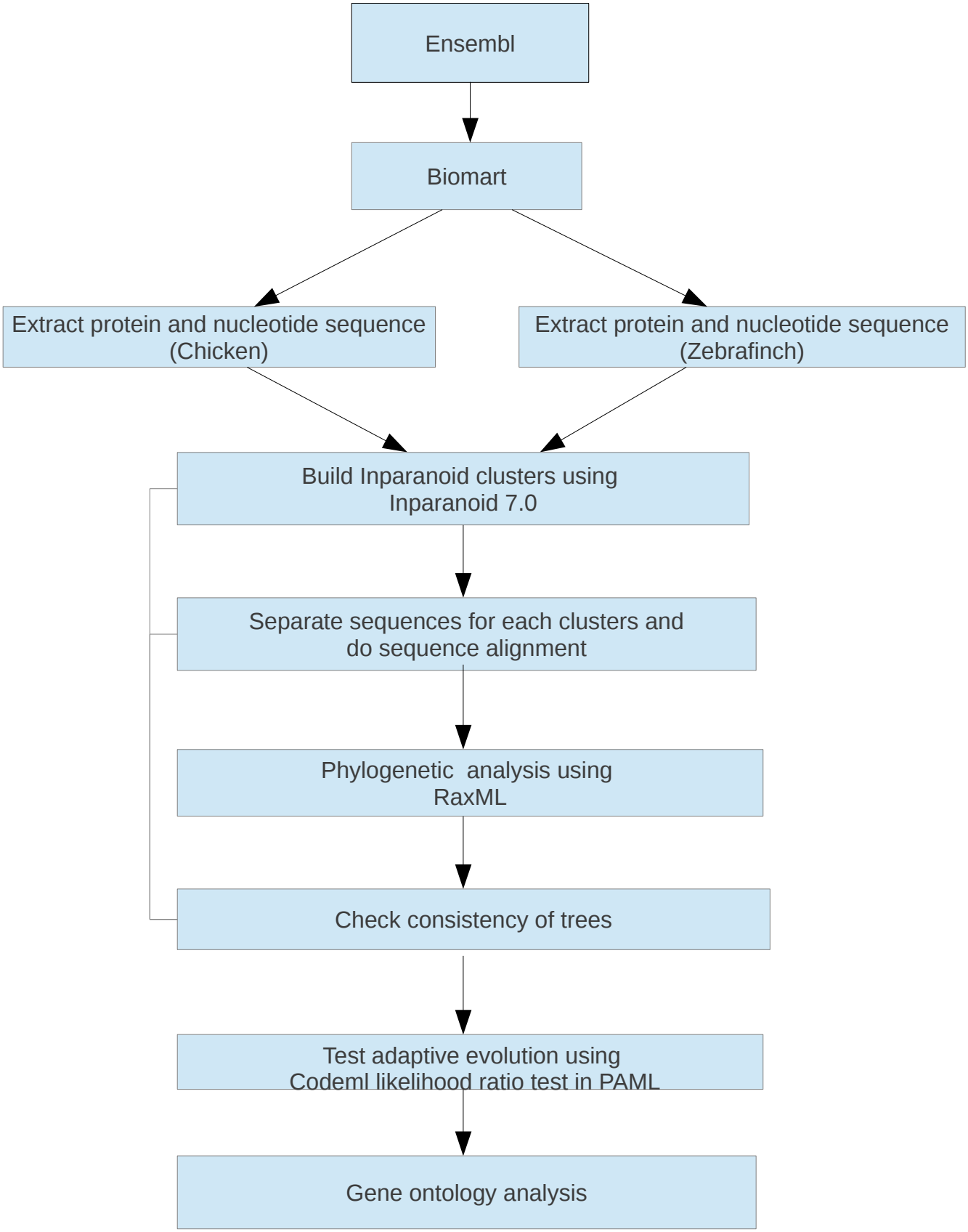

Figure 4: Flow chart showing different steps of the project

3 Results and discussion

Nam et al. conducted a study [2] on rapidly evolving genes and GO terms which were enriched in avian lineage. Another study on avian genes was conducted by Axelsson et al. [17]. In this study, the dataset was limited to genes expressed in zebrafinch brain to test for positive selection. Neither of these studies has considered whether the duplication increases the likeliness of adaptive evolution [2][17]. They were more interested in finding adaptively evolved genes.

The current study was carried out to test whether the duplication makes adaptive evolution more likely. Moreover, we have looked for categories of genes which are more enriched in the case of duplicated genes.

A proportion test was performed to check if the proportion of clusters containing duplicated genes showing significant adaptive evolution is different from the corresponding proportion among clusters containing non-duplicated genes. We counted the clusters containing duplicated genes and the non-duplicated genes which were adaptively evolved, as well as the clusters containing duplicated genes and non duplicated genes which were not adaptively evolved. Then the proportion test was applied on these gene clusters using Fisher's Exact Test for proportions as shown in the table 4.

3.1 Proportion test

Null hypothesis: Proportion of clusters where we find adaptive evolution/positive selection is same between duplicated and non duplicated clusters

Alternate hypothesis: Duplicated clusters in which we find adaptive evolution/positive selection has higher significance

Criterion: Reject null hypothesis if value of P is less than 0.05

Table 4: Proportion test between duplicated and non-duplicated gene clusters

Genes	Positive selection	No Selection
Duplicated	14	55
non-duplicated	891	9806

Result: P=0.001<0.05 which is less than 0.05, so we reject the null hypothesis and accept alternate hypothesis.

The results clearly show that clusters containing duplicated genes which are adaptively evolved have higher likelihood to be adaptively evolved then non duplicated genes clusters which are adaptively evolved. So, we can say that duplication make adaptive evolution more likely.

3.2 GO term analysis

GO term analysis was done on the duplicated genes which were found to be evolved adaptively in either species to see whether adaptive evolution occur in some gene categories more often than other. GO terms which are enriched in either species are analyzed by doing three different comparisons as shown in Table 3. Comparison 3 is shown in table 5 and table 6 in this section while comparison 1 and comparison 2 is shown in supplement section in Table S1 and Table S2 for

chicken as well as Table S3 and Table S4 for zebrafinch.
Chicken Enriched GO terms in adaptively duplicated clusters and their functions, ontology and number of genes which contain that GO term are shown in this table 5.

Table 5: GO terms enriched in chicken

No.	Go term	Function	Ontology	Count of genes	P values
1	GO:0022607	Cellular component assembly	Biological process	538	0.002446847
2	GO:0071844	Cellular Component assembly at cellular level	Biological process	404	0.000563643
3	GO:0044085	Cellular component biogenesis	Biological process	581	0.00358494
4	GO:0034622	Cellular macromolecular complex assembly	Biological process	235	3.741492e-05
5	GO:0034621	Cellular macromolecular complex subunit organization	Biological process	274	8.62231e-05
6	GO:0031497	Chromatin assembly	Biological process	63	1.539628e-08
7	GO:0006333	Chromatin assembly or disassembly	Biological process	86	1.140196e-07
8	GO:0006325	Chromatin organization	Biological process	219	3.624394e-05
9	GO:0051276	Chromosome organization	Biological process	293	0.0001698565
10	GO:0071103	DNA conformation change	Biological process	101	3.054422e-07
11	GO:0006323	DNA packaging	Biological process	76	4.344468e-08
12	GO:0046629	Gamma-delta T cell activation	Biological process	4	0.005009961
13	GO:0042492	Gamma-delta T cell differentiation	Biological process	4	0.005009961
14	GO:0065003	Macromolecular complex assembly	Biological process	379	0.0005089954
15	GO:0043933	Macromolecular	Biological	419	0.0008274468

		complex subunit organization	process		
16	GO:0006334	Nucleosome assembly	Biological process	59	7.570708e-09
17	GO:0034728	Nucleosome organization	Biological process	63	1.301503e-08
18	GO:0065004	Protein DNA complex assembly	Biological process	66	2.119439e-08
19	GO:0071824	Protein DNA complex subunit organization	Biological process	67	2.468038e-08
20	GO:0000785	Chromatin	Cellular component	141	3.182619e-06
21	GO:0044427	Chromosomal part	Cellular component	255	8.398534e-05
22	GO:0005694	Chromosome	Cellular component	296	0.0001856047
23	GO:0000786	Nucleosome	Cellular component	54	3.298866e-09
24	GO:0032993	Protein DNA complex	Cellular component	70	2.86073e-08
25	GO:0008519	Ammonium transmembrane transporter activity	Molecular function	5	3.685538e-05
26	GO:0003677	DNA binding	Molecular function	1145	0.0008145568
27	GO:0003676	Nucleic acid binding	Molecular function	1828	0.004881543
28	GO:0015101	Organic cation transmembrane transporter activity	Molecular function	8	9.184323e-05

Zebrafinch Enriched GO terms in adaptively duplicated clusters and their functions, ontology and number of genes which contain that GO term are shown in this table 6.

Table 6: GO terms enriched in zebrafinch

No.	**Go term**	**Function**	**Ontology**	**Count of genes**	**P values**
1	GO:0009310	Amine catabolic process	Biological process	42	0.004340588
2	GO:0046395	Carboxylic acid catabolic process	Biological process	63	0.01045242
3	GO:0009063	Cellular aminoacid	Biological process	38	0.003865654

		catabolic process			
4	GO:0022607	Cellular component assembly	Biological process	553	5.967603e-05
5	GO:0071844	Cellular component assembly at cellular level	Biological process	421	7.424181e-06
6	GO:0044085	Cellular component biogenesis	Biological process	593	0.0001021177
7	GO:0071842	Cellular component organization at cellular level	Biological process	1030	0.002724672
8	GO:0071841	Cellular component organization or biogenesis at cellular level	Biological process	1067	0.003438427
9	GO:0034622	Cellular macromolecular complex assembly	Biological process	269	2.941981e-07
10	GO:0034621	Cellular macromolecular complex subunit organization	Biological process	298	6.558729e-07
11	GO:0031497	Chromatin assembly	Biological process	115	6.160244e-11
12	GO:0006333	Chromatin assembly or disassembly	Biological process	136	3.149409e-10
13	GO:0006325	Chromatin organization	Biological process	250	1.118242e-07
14	GO:0051276	Chromosome organization	Biological process	312	8.489771e-07
15	GO:0071103	DNA conformation change	Biological process	146	6.269873e-10
16	GO:0006323	DNA packaging	Biological process	123	9.624519e-11
17	GO:0006548	Histidine catabolic process	Biological process	11	0.0002751862
18	GO:0009077	Histidine family aminoacid catabolic process	Biological process	11	GO:0009077
19	GO:0009075	Histidine family	Biological process	11	0.0002751862

		aminoacid metabolic process			
20	GO:0006547	Histidine metabolic process	Biological process	11	0.0002751862
21	GO:0065003	Macromolecular complex assembly	Biological process	407	8.625279e-06
22	GO:0043933	Macromolecular complex subunit organization	Biological process	438	1.449902e-05
23	GO:0006334	Nucleosome assembly	Biological process	111	3.233652e-11
24	GO:0034728	Nucleosome organization	Biological process	113	3.82115e-11
25	GO:0006996	Organelle organization	Biological process	771	0.0005375451
26	GO:0016054	Organic acid catabolic process	Biological process	63	0.01045242
27	GO:0065004	Protein DNA complex assembly	Biological process	116	7.171607e-11
28	GO:0071824	Protein DNA complex subunit organization	Biological process	116	7.171607e-11
29	GO:0007606	Sensory perception of chemical stimulus	Biological process	16	0.0005017895
30	GO:0050909	Sensory perception of taste	Biological process	10	0.0001611048
31	GO:0000785	Chromatin	Cellular component	191	1.284859e-08
32	GO:0044427	Chromosomal part	Cellular component	289	5.011046e-07
33	GO:0005694	Chromosome	Cellular component	323	1.206037e-06
34	GO:0043232	Intracellular non membrane bounded organelle	Cellular component	1360	0.00795994
35	GO:0043228	Non-membrane bounded organelle	Cellular component	1360	0.00795994
36	GO:0000786	Nucleosome	Cellular component	106	1.586729e-11
37	GO:0032993	Protein-DNA complex	Cellular component	121	8.321273e-11
38	GO:0016880	Acid ammonia (or amide) ligase activity	Molecular function	6	7.699364e-05
39	GO:0016211	Ammonia ligase activity	Molecular function	6	7.699364e-05
40	GO:0016841	Ammonia lyase activity	Molecular function	3	2.318154e-05
41	GO:0016840	Carbon nitrogen lyase activity	Molecular function	7	7.699364e-05

42	GO:0003677	DNA binding	Molecular function	1358	0.0003137298
43	GO:0003676	Nucleic acid binding	Molecular function	2081	0.007794015
44	GO:0005044	Scavenger receptor activity	Molecular function	30	0.0009057998
45	GO:0004867	Serine-type endopeptidase inhibitor activity	Molecular function	73	0.00709396

We got enriched GO terms by comparing the genes in the adaptively evolved duplicated gene cluster against nonadaptive and non-duplicated gene clusters. By looking at the function of GO terms and their graphs in chicken, we found biological process GO terms more prevalent that are involved in chromatin/cellular component assembly, chromatin/cellular component organization and DNA related biological terms. Molecular function specific terms such as ion transport activity are inconclusive due to few genes and have $p > 10^{-5}$ and we also found few general molecular function terms such as DNA and nucleic acid binding with greater number of genes. Cellular component terms such as chromatin, chromosome and nucleosome related terms are more prevalent.

In zebrafinch we have a similar biological process and cellular component GO terms which we have seen in chicken. Molecular fuction terms are inconclusive due to few genes except some general terms such as DNA and nucleic acid binding which have greater number of genes. We also found sensory perception terms enriched in zebrafinch adaptively duplicated gene clusters but they contain very few number of genes and $p > 10^{-5}$. So, they are not really trustworthy.

If we compare our study on gene enrichment analysis with the previous groups studies [2][17]. There are similarities in these studies in term of studying overrepresented GO terms in positively selected genes in chicken and zebrafinch. The only difference is that, we have also considered duplicated genes in our comparison and we believe that duplication make adaptive evolution more likely while they haven't took duplication into account while looking at GO terms in positively selected genes. Calcium ion binding and extracellular matrix are the overrepresented in ancestral birds which is similar to our comparison 1. Nam et al. [2] have found GO terms which are positively selected in chicken and zebrafinch which did not show up in our analysis. Nam et al. [2] have found terms related to neurological process which are directly related to song behaviour in zebrafinch but we found terms which are connected to neurological process but they are involved in sensory perception which I don't think have any connection with song behaviour in birds but that might be important for some other important behaviour connected with neurological system.

We have also looked whether we have any difference in terms which we got from adaptively non-duplicated genes and adaptively duplicated genes in chicken and zebrafinch. Our analysis of terms enriched in adaptively evolved duplicated genes yields an entirely different set of terms than analysis of terms enriched in adaptively non-duplicated genes.

4 Conclusions

From this study we concluded that duplications make adaptive evolution more likely. This answered one of our initial questions, i.e. does duplication make adaptive evolution more likely?

We also tried to look whether evolution occur in some gene categories more often over others during bird evolution. We thus got GO terms which are overrepresented in case of adaptively evolved duplicated gene clusters. Our analyses of adaptively evolved duplicated gene clusters revieled the same GO terms which earlier studies done by others have found, such as cell adhesion, cytoskeleton, calcium ion binding and terms related to the extracellular matrix. In both chicken and zebrafinch, we found similar biological processes and cellular components terms enriched while the exact molecular function can still not be concluded due to the fact that we have found enriched terms with few number of genes and $p>10^{-5}$.

5 Acknowledgements

I would like to thank everyone who helped me during this project. I would like to thank my supervisor, Erik Sonnhammer, for accepting me to work in his group on this project in SBC. I would like to thank him for his guidance and giving me enough time to accomplish my aims for this project.

I would like to thank my co-supervisor, Kristoffer Forslund, for his guidance and feedback throughout the project which helped me to finish it well.

I would like to thank Andreas Tjarnberg and Erik Sjolund for their guidance and helping me in developing programming skills and solving my computer problems.

I would like to thank my previous supervisor and co-supervisor, Dr Raheel Qamar and Maleeha Azam, for their guidance.

I would like to thank my friend Moeen Riaz for his guidance and help when I was applying for higher studies in Sweden, It's because of him that now I am finishing my MSc Bioinformatics from Uppsala University.

I would like to thank my parents for their support and guidance in every step of my life.

6 References

1. Swannson WJ. 2003. Adaptive evolution of genes and gene families. Elsevier. **13**: 617-622.

2. Nam K, Mugal C, Nabholz B, Schielzeth H, Wolf JBW , Backstom N, Kunstner A, Balakrishnan CN, Heger A, Ponting CP, Clayton DF, Ellegren H. 2010. Molecular evolution of genes in avian genomes. Genome Biology. **11**:R68.

3. Warren WC, Clayton DF, Ellegren H, Arnold AP, Hillier LW, Kunstner A, Searle S, White S, Vilella AJ, Fairley S, Heger A, Kong L, Ponting CP, jarvis ED, Mello CV, Minx P, Lovell P, Velho TAF, Ferris M, Balakrishnan CN, Sinha S, Blatti C, London SE, Li Y, Lin Y, George J, Sweedler J, Southey B, Gunaratne P, Watson M, Nam K, Backstrom N, Smeds L, Nabholz B, Itoh Y, Whitney O, Pfenning AR, Howard J, Volker M, Skinner BM, Griffin DK, Ye L, Mclaren WM, Flicek P, Quesada V, Velasco G, Lopez-otin C, Puente XS, Olender T, Lancet D, Smit AFA, Hubley R, Konkel MK, Walker JA, Batzer MA, GU W, Pollock DD, Chen L, Cheng Z, Eichler EE, Stapley J, Slate J, Ekblom R, Birkhead T, Burke T, Burt D, Scharff C, Adam I, Richard H, Sultan M, Soldatov A, Lehrach H, Edwards SV, Yang SP, Li X, Graves T, Fulton L, Nelson J, Chinwalla A, Hou S, Mardis ER, Wilson RK. 2010. The genome of a songbird. Nature. **464**: 757-762

4. International Chicken Genome Sequencing Consortium. 2004. Sequence and comparative analysis of the chicken genome provide unique perspectives on vertebrate evolution. Nature. **432**(7018):695-716.

5. Zhang J. 2003. Evolution by gene duplication: an update. Elsevier. **18**: 292-298

6. Stamatakis A, Ludwig T, Meier H. 2005. RaxML-III: a fast program for maximum likelihood-based inference of large phylogenetic trees. Bioinformatics **21**: 456-463

7. Price MN, Dehal PS, Arkin AP. 2010. FastTree 2 – Approximately Maximum-Likelihood Trees for Large Alignments. PloS one, **5**: e9490

8. Yang Z, Goldman N, Friday A. Comparison of Models for Nucleotide substitution used in Maximum-Likelihood Phylogenetic Estimation. Molecular Biology and Evolution **11**: 316-324

9. Yang Z. A program package for phylogenetic analysis by maximum likelihood. Cabios **13**: 555-556

10. Ashburner M, Ball CA, Blake JA, Botstein D, Butler H, Cherry JM, Davis AP, Dolinski K, Dwight SS, Eppig JT, Harris MA, Hill DP, Issel-Tarver L, Kasarskis A, Lewis S, Matese JC, Richardson JE, Ringwald M, Rubin GM, Sherlock G. 2000. Gene Ontology:tool for the unification of biology. The gene ontology consortium. Nat Genet **25**: 25-29

11. Smedley D, Haider S, Ballester B, Holland R, London D, Thorisson G, Kasprzyk A. 2009.

BioMart--biological queries made easy. BMC Genomics. **10**:22.

12. Östlund G, Schmitt T, Forslund K, Köstler T, Messina DN, Roopra S, Frings O, Sonnhammer ELL. et al. 2010. InParanoid 7: New algorithms and tools for eukaryotic orthology analysis. Nucleic Acids Research **38**: D196-D203

13. Benjamini Y, Hochberg Y. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society, Series B (Methodological) **57**: 289–300

14. Lomax J. 2005. Get ready to GO! A biologist's guide to the Gene Ontology. Brief Bioinform **6**: 298-304.

15. Lassmann T, Sonnhammer ELL. 2005. Kalign-an accurate and fast multiple sequence alignment algorithm. BMC Bioinformatics **6**:298

16. Castresana J. Selection of conserved Blocks from Multiple Alignments for their use in Phylogenetic Analysis. Mol Biol.Evol **17**: 540-552

17. Axelsson E, Rosenberg LH, Brandstrom M, Zwahlen M, Clayton DF, Ellegren H. 2008. Natural selection in avian protein-coding genes expressed in brain Molecular ecology **17**: 3008-3017

18. Rokas A. 2011. Phylogenetic Analysis of Protein Sequence Data Using the Randomized Axelerated Maximum Likelihood (RAXML) Program. Current protocols in Moleular Biology, doi: 10.1002/0471142727.mb1911s96

7 Supplements

Table S1: GO terms enriched in chicken (Comparison 1)

No.	Go term	Function	Ontology
1	GO:0051056	Regulation of small GTPase-mediated signal transduction	Biological process
2	GO:0005604	Basement membrane	Cellular component
3	GO:0005581	Collagen	Cellular component
4	GO:0031012	Extracellular matrix	Cellular component
5	GO:0005578	Proteinaceous extracellular matrix	Cellular component
6	GO:0044420	Extracellular matrix part	Cellular component
7	GO:0005201	Extracellular matrix structure constituent	Molecular function
8	GO:0030695	GTPase regulator activity	Molecular function
9	GO:0060589	Nucleoside-triphosphate regulator activity	Molecular function
10	GO:0005088	Ras guanyl-nucleotide exchange factor activity	Molecular function
11	GO:0005089	Rho guanyl-nucleotide exchange factor activity	Molecular function
12	GO:0065083	Small GTPase regulator activity	Molecular function

Table S2: GO terms enriched in chicken (Comparison 2)

No.	Go term	Function	Ontology
1	GO:0031497	Chromatin assembly	Biological process
2	GO:0006333	Chromatin assembly and disassembly	Biological process
3	GO:0071103	DNA conformation change	Biological process
4	GO:0006323	DNA packaging	Biological process
5	GO:0006334	Nucleosome assembly	Biological process
6	GO:0034728	Nucleosome organization	Biological process

7	GO:0065004	Protein-DNA complex assembly	Biological process
8	GO:0071824	Protein-DNA complex subunit organization	Biological process
9	GO:0005615	Extracellular space	Cellular component
10	GO:0000786	Nucleosome	Cellular component
11	GO:0032993	Protein-DNA complex	Cellular component
12	GO:0008519	Ammonium transmembrane transporter activity	Molecular function
13	GO:0008009	Chemokine activity	Molecular function
14	GO:0042379	Chemokine receptor binding	Molecular function
15	GO:0005125	Cytokine activity	Molecular function
16	GO:0005126	Cytokine receptor binding	Molecular function
17	GO:0001664	G-protein coupled receptor binding	Molecular function
18	GO:0005102	Receptor binding	Molecular function
19	GO:0008518	Reduced folate activity	Molecular function
20	GO:0017171	Serine Hydrolase activity	Molecular function
21	GO:0004252	Serine type endopeptidase activity	Molecular function
22	GO:0008236	Serine-type peptidase activity	Molecular function
23	GO:0008146	Sulfotransferase activity	Molecular function
24	GO:0016782	Transferase activity, transferring sulphur-containing groups	Molecular function

Table S3: GO terms enriched in zebrafinch (Comparison 1)

No.	**Go term**	**Function**	**Ontology**
1	GO:0071103	DNA conformation change	Biological process
2	GO:0051056	Regulation of small GTPase mediated signal transduction	Biological process
3	GO:0005581	Collagen	Cellular component
4	GO:0031012	Extracellular matrix	Cellular component
5	GO:0000786	Nucleosome	Cellular component

6	GO:0032993	Protein-DNA complex	Cellular component
7	GO:0005578	Proteinaceous extracellular matrix	Cellular component
8	GO:0015301	Anion anion Antiporter activity	Molecular function
9	GO:0030234	Enzyme regulator activity	Molecular function
10	GO:0005201	Extracellular matrix structural constituent	Molecular function
11	GO:0030695	GTPase regulator activity	Molecular function
12	GO:0005452	Inorganic anion exchanger activity	Molecular function
13	GO:0060589	Nucleotide triphosphate regulator activity	Molecular function
14	GO:0048407	Platelet-derived growth factor binding	Molecular function
15	GO:0005089	Rho guanyl-nucleotide exchange factor activity	Molecular function

Table S4: GO terms enriched in zebrafinch (Comparison 2)

No.	**Go term**	**Function**	**Ontology**
1	GO:0006526	Arginine biosynthetic process	Biological process
2	GO:0031497	Chromatin assembly	Biological process
3	GO:0006333	Chromatin assembly or disassembly	Biological process
4	GO:0071103	DNA conformation change	Biological process
5	GO:0006323	DNA packaging	Biological process
6	GO:0006334	Nucleosome assembly	Biological process
7	GO:0034728	Nucleosome organization	Biological process
8	GO:0006591	Ornithine metabolic process	Biological process
9	GO:0006813	Potassium ion transport	Biological process
10	GO:0065004	Protein-DNA complex assembly	Biological process
11	GO:0071824	Protein-DNA complex subunit organization	Biological process
12	GO:0006465	Signal peptide processing	Biological process
13	GO:0034703	Cation channel complex	Cellular component
14	GO:0000785	Chromatin	Cellular component
15	GO:0005795	Golgi complex	Cellular component

16	GO:0034702	Ion channel complex	Cellular component
17	GO:0000786	Nucleosome	Cellular component
18	GO:0034705	Potassium channel complex	Cellular component
19	GO:0032993	Protein-DNA complex	Cellular component
20	GO:0005787	Signal peptidase complex	Cellular component
21	GO:0008076	Voltage-gated potassium channel complex	Cellular component
22	GO:0016880	Acid-ammonia (or amide) ligase activity	Molecular function
23	GO:0004030	Aldehyde dehydrogenase[NAD(P)+] activity	Molecular function
24	GO:0016842	Amidine-lyase activity	Molecular function
25	GO:0016211	Ammonia ligase activity	Molecular function
26	GO:0016841	Ammonia lyase activity	Molecular function
27	GO:0016840	Carbon nitrogen lyase activity	Molecular function
28	GO:0008009	Chemokine activity	Molecular function
29	GO:0042379	Chemokine receptor binding	Molecular function
30	GO:0005126	Cytokine receptor binding	Molecular function
31	GO:0008378	Galactosyltransferase activity	Molecular function
32	GO:0022836	Gated channel activity	Molecular function
33	GO:0005267	Potassium channel activity	Molecular function
34	GO:0004800	Thyroxine 5-deiodinase activity	Molecular function
35	GO:0022843	Voltage gated cation channel activity	Molecular function
36	GO:0022832	Voltage gated channel activity	Molecular function
37	GO:0005244	Voltage gated ion channel activity	Molecular function
38	GO:0005249	Voltage gated potassium channel activity	Molecular function

Control file:

codeml:

Phylogenetic Analysis by Maximum Likelihood (PAML) has its own control file for its execution [9][10]. When the codeml application is run, it will look for the control file “codeml.ctl” for execution.
For this project the control file parameters are changed based on which criteria we want to use. The first two lines of the file indicate the names of the sequence and tree files. These lines are changed by a perl script for each gene cluster analysis for adaptive evolution. The control file is given the same file prefix as these files, followed by “.ctl”. The remainder of the file contents is copied from the template file. In file we have changed model and NSsites parameters before applying the Perl script to change the alignment and tree file. The template was built from the examples provided with the download of the software:

seqfile = xxx.phy * alignment file
treefile = xxx.phy * tree file

outfile = mlc * main result file name

noisy = 9 * 0,1,2,3,9: how much rubbish on the screen

runmode = 0 * 0: user tree; 1: semi-automatic; 2: automatic

seqtype = 1 * 1:codons; 2:AAs; 3:codons-->AAs

CodonFreq = 2 * 0:1/61 each, 1:F1X4, 2:F3X4, 3:codon table

* ndata = 10

clock = 0 * 0:no clock, 1:clock; 2:local clock; 3:CombinedAnalysis

aaDist = 0 * 0:equal, +:geometric; -:linear, 1-6:G1974,Miyata,c,p,v,a

aaRatefile = dat/jones.dat * only used for aa seqs with model=empirical(_F)

model = 0

NSsites = 1 2 * 0:one w;1:neutral;2:selection; 3:discrete;4:freqs;

icode = 0 * 0:universal code; 1:mammalian mt; 2-10:see below

Mgene = 0

fix_kappa = 0 * 1: kappa fixed, 0: kappa to be estimated

kappa = 2 * initial or fixed kappa

fix_omega = 0 * 1: omega or omega_1 fixed, 0: estimate

omega = .4 * initial or fixed omega, for codons or codon-based AAs

fix_alpha = 1 * 0: estimate gamma shape parameter; 1: fix it at alpha

alpha = 0. * initial or fixed alpha, 0:infinity (constant rate)

Malpha = 0 * different alphas for genes

ncatG = 8 * # of categories in dG of NSsites models

getSE = 0 * 0: don't want them, 1: want S.E.s of estimates

RateAncestor = 1 * (0,1,2): rates (alpha>0) or ancestral states (1 or 2)

Small_Diff = .5e-6

cleandata = 1 * remove sites with ambiguity data (1:yes, 0:no)?

* fix_blength = -1 * 0: ignore, -1: random, 1: initial, 2: fixed

method = 0 * Optimization method 0: simultaneous; 1: one branch a time