Universitat Oberta de Catalunya

uoc.edu

Classification between odorant and gustatory receptors: Supervised learning applied to arthropod proteins

Autor: Félix Francisco Enríquez Romero
Máster universitario en Bioinformática y Bioestadística
Área 3 - Evolución Molecular

Director externo (UB): Dr. Alejandro Sánchez-Gracia
Consultora UOC: Dra. Dorcas Orengo Ferriz
Profesora responsable de la asignatura: Dra. Laura Calvet Liñán

Fecha de entrega: 01/2022

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Classification between odorant and gustatory receptors: Supervised learning applied to arthropods proteins*
Nombre del autor:	*Félix Francisco Enríquez Romero*
Nombre del consultor/a:	*Dorcas Orengo Ferriz*
Nombre del PRA:	*Laura Calvet Liñán*
Fecha de entrega (mm/aaaa):	*01/2022*
Titulación:	*Máster universitario en Bioinformática y Bioestadística UOC-UB*
Área del Trabajo Final:	*Área 3 - Evolución Molecular*
Idioma del trabajo:	*Inglés*
Número de créditos:	15
Palabras clave	*odorant receptor, gustatory receptor, supervised learning*

Resumen del Trabajo (máximo 250 palabras): *Con la finalidad, contexto de aplicación, metodología, resultados y conclusiones del trabajo.*

La caracterización funcional de proteínas es difícil de conseguir mediante sistemas automáticos basados en alineamientos de secuencias. Estos sistemas no han encontrado evidencia de que receptores olfativos estén presentes en artrópodos no insectos. En este proyecto se desarrolla un sistema basado en aprendizaje supervisado que primeramente sea capaz de clasificar la funcionalidad de proteínas provenientes de insectos en receptores de tipo olfativo y receptores de tipo gustativo y que posteriormente pueda identificar en secuencias de otros artrópodos (anotadas como receptores gustativos por sistemas de alineamiento de secuencias) secuencias que hayan sufrido una divergencia funcional similar a la ocurrida en los insectos alados hace unos 440 millones de años. Las secuencias que conforman el dataset se obtuvieron de UniProtKB, tres métodos distintos fueron usados para codificar las secuencias y poder entrenar tres redes neuronales artificiales. Los resultados obtenidos confirman la eficacia del modelo desarrollado con un AUC, precisión y F1 de 0.911, 0.971 y 0.926, respectivamente. Se obtuvo un listado de secuencias candidatas a tener una funcionalidad parecida a receptores olfativos en artrópodos no insectos. El análisis posterior de la funcionalidad de estas secuencias podría corroborar las predicciones del modelo. El hallazgo de la presencia de proteínas con funcionalidad parecida a los receptores olfativos sería indicativo de una posible convergencia evolutiva.

Abstract (in English, 250 words or less):

Functional characterisation of proteins is limited and difficult to achieve using automatic systems based on sequence alignments which have not found evidence of the presence of odorant receptors in other arthropods than insects. In this project, a supervised learning based system is developed in order to first classify insect protein sequences functionality as odorant or gustatory receptors and second, identify in non-insect arthropod sequences (annotated as gustatory receptors by alignment based systems) potential sequences that suffered a similar functional divergence as it happened in insects around 440 million years ago. Dataset sequences were obtained from UniProtKB, three different encoding methods were used to train three artificial neural networks. The results obtained confirm the efficiency of the developed model with AUC, precision and F1 score of 0.911, 0.971 and 0.926, respectively. A list of candidate sequences to have a functionality similar to that of odorant receptors in non-insect arthropods was obtained. Downstream analysis regarding functionality of these sequences should be done to corroborate the model predictions. Findings regarding the presence of proteins with similar functionality to odorant receptors in other arthropods would indicate an extraordinary evolutionary convergence.

Contents

List of figures

List of tables

1 Abstract

Functional characterisation of proteins is limited and difficult to achieve using automatic systems based on sequence alignments which have not found evidence of the presence of odorant receptors in other arthropods than insects. In this project, a supervised learning based system is developed in order to first classify insect protein sequences functionality as odorant or gustatory receptors and second, identify in non-insect arthropod sequences (annotated as gustatory receptors by alignment based systems) potential sequences that suffered a similar functional divergence as it happened in insects around 440 million years ago. Dataset sequences were obtained from UniProtKB, three different encoding methods were used to train three artificial neural networks. The results obtained confirm the efficiency of the developed model with AUC, precision and F1 score of 0.911, 0.971 and 0.926, respectively. A list of candidate sequences to have a functionality similar to that of odorant receptors in non-insect arthropods was obtained. Downstream analysis regarding functionality of these sequences should be done to corroborate the model predictions. Findings regarding the presence of proteins with similar functionality to odorant receptors in other arthropods would indicate an extraordinary evolutionary convergence.

2 Introduction

2.1 Context and justification

Chemosensory receptors are trans-membrane proteins present in all organisms whose function is to detect molecules from a broad range of chemicals; consequently, we can find different receptor subfamilies with different chemoreceptor functions in the organism. In insects, there are two main chemoreceptor subfamilies: gustatory (GRs) and odorant (ORs) receptors, which jointly with the family ionotropic receptors (IRs), are the responsible of chemoperception in these animals [1].

Insect chemoreceptors are not homologous to their vertebrate counterparts. Furthermore, recent comparative genomic studies would indicate that the ORs originated from a lineage of GRs in winged insects, likely as a consequence of the functional and structural divergence associated with flight adaptation in this group of species. This fact would explain why this class of OR is not found in the genome of any arthropod other than insect [1; 2]. However, it is still unknown whether similar adaptations to terrestrial environments in other arthropod lineages resulted in the invention of proteins with molecular properties similar to those of insect ORs, which have not been detected so far using sequence similarity-based searches.

Currently, a vast amount of genomic data is available from sequencing projects, but the accurate identification and functional characterisation of proteins is limited and difficult to achieve using automatic methods, which often need further extensive human-based curations. In this project, a supervised learning-based system to detect functional divergence patterns encoded in the amino acid sequences of insect ORs and GRs is proposed. The system is conceived as an alternative tool to sequence similarity-based searches, which might be not powerful enough to detect functional shifts in highly diverged chemoreceptors similar to those occurring in flying insects [3].

The proposed system aims to classify chemoreceptors identified in a number of available genome sequences of non-insect species, and that are expressed in the putative chemosensory organs of this species, as OR or GR-like proteins. Results can help to identify potential receptors for volatile hydrophobic molecules derived from GRs in arthropod lineages other than insects (replicating what occurred in insects), and therefore, evolutionary convergences during the adaptation to land in this group of organisms.

2.2 Objectives

General objectives

1.To create a classifier modelled with annotated insect sequences from the Uniprot database [4] applying supervised Machine Learning (ML) techniques that allows differentiating odorant and gustatory receptors based on their amino acid sequences.

2.To develop a Python package [5] publicly available in GitHub [6], so researchers can download and use the classifier.

Specific objectives

1.To create a classifier modelled with annotated insect sequences from the Uniprot database [4] applying supervised Machine Learning (ML) techniques that allows differentiating odorant and gustatory receptors based on their amino acid sequences.

1.1.Obtain a reliable set of insect sequences from the Uniprot database.

1.2.Preprocess the sequences.

1.3.Design the classifier.

1.4.Train, evaluate and test the classifier using the preprocessed dataset.

1.5.Identify sequences annotated as GR in non-insect arthropods with functional shifts similar to those that occurred in OR of insects.

1.6.Analyse gene expression data from OR candidate sequences.

2.To develop a Python package publicly available in GitHub, so researchers can download and use the classifier.

2.1.Allow software free distribution.

2.2.Ensure software usability.

2.3 Approach and method to follow

Quality of input data is one of the most limiting factors in terms of Artificial Intelligence (AI) models performance. Thus, it is crucial for this project to obtain high quality insect OR and GR sequences.

The Uniprot database will be used to obtain curated and non-curated sequences from ORs and GRs from insects. Sequences will be filtered using InterProScan so that sequences that do not show or show less than 60% of the receptor's protein domain known for the receptor will be discarded.

At this point, sequences must be encoded so that most of the functional properties associated with each amino acid are included. Given the vast number of potential characteristics available, dimensionality reduction techniques will be applied so compressed data can be then used for the model.

Next step to obtain high quality performance of the model is the hyper parameter tuning and validation process, which allows an estimator to fit the input data avoiding overfitting. In this project, non-curated sequences will represent the training, validation and test datasets, the curated sequences will be used as a curated set to test the model since they are quite accurate (although scarce, see below).

2.4 Work plan

Tasks

1.1.Specific objective: Obtain a reliable set of insect sequences from the Uniprot database:

- 1.1.1. OR and GR sequences and metadata extraction from Uniprot database.
- 1.1.2. FASTA files parsing.
- 1.1.3. Sequence filtering using InterProScan.

1.2.Specific objective: Preprocess sequences:

- 1.2.1. Sequence encoding with amino acid physicochemical properties.
- 1.2.2. Data characterization.
- 1.2.3. Dimensionality reduction of the dataset.
- 1.2.4. Data partition into train, validation and test sets.

1.3.Specific objective: Design the classifier:

- 1.3.1. Classifiers and meta classifiers exploration.
- 1.3.2. Classifier selection.

1.4.Specific objective: Train, evaluate and test the classifier using the preprocessed dataset:

- 1.4.1. Hyper parameter search.
- 1.4.2. Model evaluation and testing.

1.5.Identify sequences annotated as GR in non-insect arthropods with functional shifts similar to those that occurred in OR of insects:

- 1.5.1. Extraction of non-insect arthropod chemoreceptors currently annotated as GR.
- 1.5.5. Pre-processing of non-insect arthropod chemoreceptors currently annotated as GR.
- 1.5.6. Prediction of potential insect OR functionally related sequences of non-insect arthropods.

1.6.Specific objective: Analyse gene expression data from sequences OR candidate sequences:

- 1.6.1. Gene ontology analysis from assays.

2.1.Specific objective: Allow software free distribution:

- 2.1.1. Project creation on GitHub.

2.2.Specific objective: Ensure software usability:

- 2.2.1. Readme document drafting.

2.2.2. Use cases scripting.

·Tasks unrelated with specific objectives:

-Work proposal drafting.

-Work planning.

-Project document drafting.

-Project document reviewing.

-Project presentation drafting (slides).

-Project defense.

Tasks duration

For each task duration is estimated, weekends are excluded.

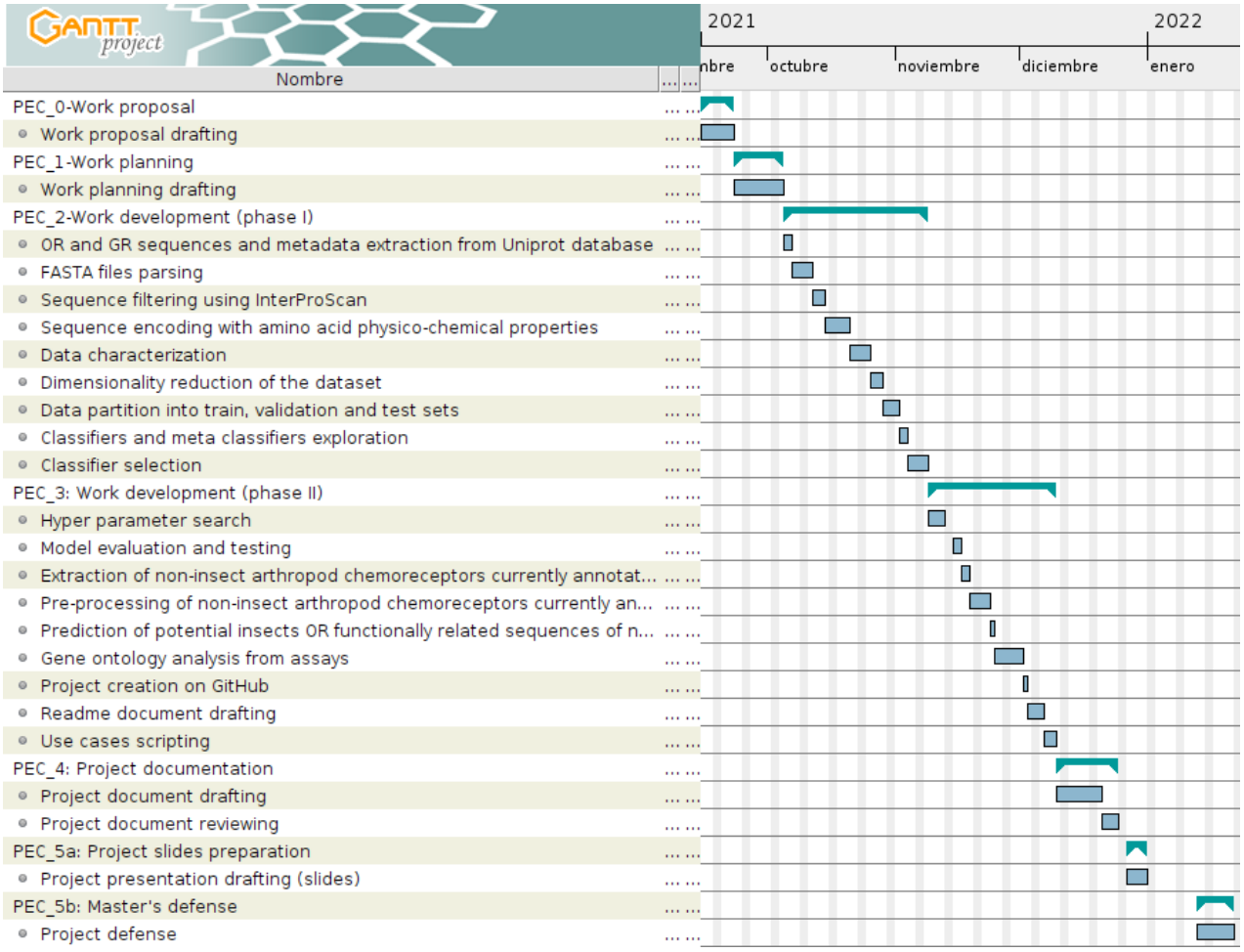
Table 1: List of tasks duration

Task name	Specific objective related	Duration (working days)	Start date	End date
PEC_0-Work proposal	-	**6**	**15/09/2021**	**22/09/2021**
Work proposal drafting	-	6	15/09/2021	22/09/2021
PEC_1-Work planning	-	**8**	**23/09/2021**	**04/10/2021**
Work planning drafting	-	8	23/09/2021	04/10/2021
PEC_2-Work development (phase I)	-	**25**	**05/10/2021**	**08/11/2021**
OR and GR sequences and metadata extraction from *Uniprot* database	1.1	2	05/10/2021	06/10/2021
FASTA files parsing	1.1	3	07/10/2021	11/10/2021
Sequence filtering using InterProScan	1.1	3	12/10/2021	14/10/2021
Sequence encoding with amino acid physicochemical properties	1.2	4	15/10/2021	20/10/2021
Data characterization	1.2	3	21/10/2021	25/10/2021
Dimensionality reduction of the dataset	1.2	3	26/10/2021	28/10/2021
Data partition into train, validation and test sets	1.2	2	29/10/2021	01/11/2021
Classifiers and meta classifiers exploration	1.3	2	02/11/2021	03/11/2021
Classifier selection	1.3	3	04/11/2021	08/11/2021
PEC_3-Work development (phase II)	-	**23**	**09/11/2021**	**09/12/2021**

Hyper parameter search	1.4	4	09/11/2021	12/11/2021
Model evaluation and testing	1.4	2	15/11/2021	16/11/2021
Extraction of non-insect arthropod chemoreceptors currently annotated as GR	1.5	2	17/11/2021	18/11/2021
Pre-processing of non-insect arthropod chemoreceptors currently annotated as GR	1.5	3	19/11/2021	23/11/2021
Prediction of potential insect OR functionally related sequences of non-insect arthropods	1.5	1	24/11/2021	24/11/2021
Gene ontology analysis from assays	1.6	5	25/11/2021	01/12/2021
Project creation on GitHub	2.1	1	02/12/2021	02/12/2021
Readme document drafting	2.2	2	03/12/2021	06/12/2021
Use cases scripting	2.2	3	07/12/2021	09/12/2021
PEC_4-Project documentation	**-**	**11**	**10/12/2021**	**24/12/2021**
Project document drafting	-	7	10/12/2021	20/12/2021
Project document reviewing	-	4	21/12/2021	24/12/2021
PEC_5a-Project slides preparation	**-**	**5**	**27/12/2021**	**31/12/2021**
Project presentation drafting (slides)	-	5	27/12/2021	31/12/2021
PEC_5b: Master's defense	**-**	**7**	**13/01/2022**	**21/01/2022**
Project defense	-	7	13/01/2022	21/01/2022

Gantt Diagram

Figure 1: Gantt Diagram



Milestones

1.PEC_0: Work proposal.
2.PEC_1: Work planning.
3.PEC_2: Work development (phase I).
4.PEC_3: Work development (phase II).
5.PEC_4: Project documentation.
6.PEC_5a: Project slides preparation.
7.PEC_5b: Master's defense.

Risks analysis

Possible risks factors:

-Storage capacity: If after encoding sequences local machine memory is not enough, a virtual machine could be an alternative by the time the project is developed.

-Non-identification of insect OR functionally related sequences in non-insect arthropods: In this case, no downstream analysis can be performed on gene expression data so specific objective 1.6 will not be accomplished. We should explore new variables to include in the model, such as 3D structure related parameters or site-specific evolutionary rates.

-Technical issues such as bugs in the code: In order to minimize this risk factor, commenting and reviewing of the code should be done at every stage of the project.

-Other factors related to personal/professional unforeseen circumstances: In this case evaluating and testing the classifier is a priority while specific objectives 1.5 and 1.6 are secondary.

2.5 Summary of products obtained

From this project we obtain:

-Related to UOC:

1)Work proposal document.

2)Work planning including tasks, calendar and Gantt chart.

3)Follow-up report Phase I.

4)Follow-up report Phase II.

5)Final thesis document.

6)Presentation of the project (slides).

-Related to the research:

1) A classifier developed in Python programming language for OR/GR sequences classification in GitHub.

2.6 Brief description of the other chapters of the report

Chapter 3: Introduces the state of the art related to protein databases, chemoreceptors and supervised learning applied to omics data.

Chapter 4: Describes in detail methods applied in the development of this project.

Chapter 5: Presents models evaluation and scores obtained.

Chapter 6: Limitations and future word of the project is discussed as well as if results meet the initial objectives of the project.

Chapter 7: List the most relevant results and conclusions derived from the research.

Chapter 8: Glossary of most relevant words and acronyms mentioned in the report.

Chapter 9: Includes all references.

3 State of the art

Success of Machine Learning (ML) models does not only depend on the algorithm used or hyper parameter setup but also on the quality of the input data and the understanding of the problem to be solved. This section begins stating the biological challenge with chemoreceptors and describing the global hub of protein knowledgebase, next a review of current methods involved in sequence encoding and in protein function prediction is done.

Animal chemosensory receptors are trans-membrane proteins (individual molecules or protein complexes) present in all organisms. Chemoreceptors trigger a cellular response when they bind ligands. They can be found in the cell surface, and can be categorised in three main groups: G-protein-linked receptors, ligand-gated ion channel receptors and enzyme-linked receptors.

The biological importance of chemoreceptors lies in the fact that they allow the organism to create an internal representation of the outside environment. Environmental cues are inputs, and organism behaviour responses are outputs of the sensory machinery that respond to these external signalling molecules. Chemicals come from different sources as they can be indicative of food proximity, potential pairing mates or threats.

Three main types of chemosensory receptors are found in arthropods: odorant (ORs) and gustatory (GRs) receptors (known as “the chemoreceptor superfamily”), and ionotropic receptors (IRs), a divergent lineage derived from ionotropic glutamate receptors (iGluR). However, ORs are exclusive of insects as they have not been found in similarity-based searches on the genomes of the remaining arthropod lineages [1; 2].

In vertebrates, all ORs are G-protein-coupled receptors, but in the case of insects, OR are ion gated channels receptors [1; 7]. In *Drosophila*, ORs are in the membrane of the odorant receptor neurons (ORNs), which are located inside hair-like structures called sensilla that cover the olfactory organs of this insect (the antennae and maxillary palps). Odorant molecules get in contact with receptors in the membrane of ORN dendrites by accessing through pores in these sensilla, often bound to an odorant binding protein (OBP) [1; 7; 8].

Unlike ORs, GR functional mechanism is less understood. It is largely unknown whether they can act as ion channels like some ORs [7]. ORs evolved from a specific lineage of GRs in terrestrial insects [1; 7; 8] likely promoting the adaptation of flying insects to new food resources and chemical communication systems based on volatile molecules [1]. In addition to *Drosophil*a, the most studied insect by large, evidence of the functional role of ORs has been found in many other species, such as for example in ants, where *Harpegnathos sp*. OR263 and OR348 are 93% identical in their sequence but only OR263 shows a high response to the queen pheromone [9].

Cost reduction of the process of sequencing has allowed the scientific community to generate an ever-increasing amount of sequences that has not or will never be experimentally characterised, creating the necessity of more accurate and robust automatic and semi-automatic classification and annotation methods [4; 10].

UniProt Knowledgebase (UniProtKB) is a freely accessible repository of protein sequences that includes functional annotation information. It contains over 219 million sequence entries belonging to 1.255.356 different species (release 2021_03). The aim of UniProtKB is to support biology and biomedicine fields by providing a complete compendium of all known protein sequence data linked to functional information about each protein [4]. Two main databases can be found in UniProtKB: Swiss-Prot and TrEMBL (Translated EMBL). Sequences in Swiss-Prot database are annotated by expert curators whilst sequences in TrEMBL are annotated by automated systems. Due to the growing amount of genomic data from sequencing projects, unreviewed sequences represent most sequences present in UniProtKB.

Expert curation is the process of reviewing and evaluating scientific literature from multiple sources that provide experimental evidence of the functional information of proteins. It is a labour-intensive process but still "the most efficient method of extracting all relevant data from a paper" [11].

Automated annotation systems use predictive models (such as hidden Markov models) trained on experimentally characterized sequences in order to annotate unreviewed sequences [10]. These systems require the presence of an ordered region of protein that can be recognized as a protein domain or a signature of family membership in a specific database of protein families [4]. InterPro is the one of the most popular and extensive of these databases and is commonly used for protein family identification and annotation. It integrates 14 different protein signature databases and every new entry to be added is examined by expert curators in advance.

UniProtKB uses InterProScan which is a software to search TrEMBL sequences against InterPro's predictive models. These models describe protein families, domains or sites and are provided by its integrated protein signature databases. The proportion of sequences in UniProtKB with at least one InterPro annotation is over 80% [10]. Unreviewed sequences in UniProtKB with missing InterPro annotation "seems to derive from intrinsically disordered (ID) protein regions" [4].

Sequences from TrEMBL database are also enriched using two complementary systems: UniRule and ARBA (Association Rule Based Annotator). UniRule is a rule-based computational annotation system (6768 rules in release 2020_04) that annotates TrEMBL proteins based on similarities to known experimentally characterized proteins, properties such as protein name and functionality are annotated by UniRule [4]. To complement annotations from UniRule, ARBA is used as a classification and annotation system. It is trained with Swiss-Prot data and generates concise annotation models. ARBA has been recently released by

UniProtKB (release 2020_04) and has replaced SAAS (Statistical Automatic Annotation System), an annotation system developed by UniProtKB [4].

Once data are retrieved from databases, sequences need to be processed in order to be used as input for ML models. When dealing with discrete data (text, categorical data, sequence data such as amino acid sequences) a numerical conversion needs to be done as a previous step. For protein amino acid sequences there exist several methods that differ in the information used to make a digital representation of each amino acid, these methods are Binary encoding, Physicochemical properties encoding, Evolution-based encoding, Structure-based encoding and Machine learning encoding [12].

One-hot encoding is one of the most widely used techniques of binary encoding. It consists in representing each amino acid as a 20-length vector where every position of the vector corresponds to one of the twenty common proteinogenic amino acids. In each vector, only the position associated with the amino acid to be encoded will be set to one. The length of the encoded sequence will be the length of the original sequence multiplied by the length of the vector used to encode. One drawback of One-hot encoding is that sparse vectors are obtained increasing model complexity.

Physicochemical properties of the amino acids that form a protein play a critical role in their structure and therefore also in their function [12]. Properties like hydrophobicity, polarity and solvation free energy are used to make a feature representation in the shape of a numerical vector for each amino acid. This encoding method achieved better results than others when doing one class classification for disease gene identification [13]. Statistical analysis of physical properties is also used in order to describe the twenty standard amino acids, Kidera et al. [14] developed a method to describe 86% of 188 physical properties of amino acids as a sum of 10 orthogonal factors.

Evolution-based encoding uses information from sequence alignments or phylogenetic trees to encode amino acids. Evolution-based methods are grouped in position-independent and position-dependent methods [12]. Position-independent methods use Position Accepted Mutation (PAM) matrices and BLOcks of Amino Acid SUbstitution Matrix (BLOSUM) matrices to encode amino acids. PAM matrices are based on observed mutations through a global alignment opposite to BLOSUM matrices that are based on highly conserved regions. "Position-dependent encodings are deduced from the multiple sequence alignments (MSAs) of target sequence" [12]. MSAs representation is used in AlphaFold as an input for protein structure prediction [15].

The structure-based encoding method uses inter-residue contact information representing the distance between all existing amino acid pairs in a protein in the shape of a two-dimensional matrix known as contact map.

There are two main approaches when applying Machine learning encoding, one is the usage of autoencoders, and another is to treat sequences as text.

Autoencoders are neural networks that work as an identity function, the aim of autoencoders is to compress data such that it can be represented in a lower dimensional space (latent space). This lower dimensional representation is used as a compressed-encoded version of the original encoding.

From the perspective of Natural Language Processing (NLP) each amino acid sequence represents a sentence and an amino acid or sub-sequence of amino acids a word. One of the most common NLP methods for encoding of sequences is bag-of-words (BoW) [16], which is a representation that describes word occurrences in a document in the shape of a matrix where columns are unique words and rows are word occurrences in each document. However, BoW results in a high dimensional representation of the data.

ML models require a fix-shape input vector [16; 17]. To fit that specification, different length sequences need to be transformed in a way that all have the same length. Zero-padding is a technique consisting in adding zeros to encoded sequences until the required length is achieved. Padding can be done in multiple ways (post-padding, pre-padding, stratified, random padding, ...) Lopez-del Rio et al. [17] measured the effect of different types of padding on model performance.

According to Gene Ontology (GO) [18] classification of proteins can be done based on its molecular function, biological processes, and cellular components [19]. Different methods to predict protein amino acid function are available and a distinction can be done between conventional methods (similarity- and probabilistic-based methods) and ML based methods.

Similarity-based methods consist in searching the query sequence in an experimentally curated protein database. Applying local alignment search tools such as BLAST (Basic Local Alignment Search Tool) the sequence with the highest alignment score is returned and its GO terms are assigned to the query sequence [20].

Probabilistic-based methods employ a functional linkage graph based on a Protein-Protein Interactions Network (PPI-Net). "The working assumption was that the probability of sharing functions between (nodes) proteins in close proximity on the graph is higher than that for nodes that are not in close proximity" [20].

From the perspective of Machine Learning, protein function prediction is a classification task. Classification is a type of supervised learning where an algorithm is trained on a labeled dataset beforehand. Classical ML algorithms for classification include random forest, K nearest neighbours or support vector machines whereas algorithms based on neural networks are considered Deep Learning (DL). DL is a branch of ML where algorithms consist of several hidden layers (deep) of nodes (neurons) that are inspired in biological neurons.

Aside from the algorithm used, DL algorithms do not require feature extraction (as adding new attributes to the dataset) given that they can discover hidden

underlying patterns in the data, this becomes a powerful feature when function similarity is a consequence of convergent evolution or when sequences with similar functionality are evolutionary distant.

To boost the predictive performance of models, hyper parameter search is a required step when building a model (once the dataset is prepared and an algorithm is chosen). Hyper parameters govern how parameters are estimated inside the algorithm. For a neural network hyper parameters include learning rate, number of hidden layers, number of units (neurons) per layer and parameters are the weights of the network (used to make predictions) that are obtained after the training process is completed. The training process consists in minimizing/maximizing a cost function that reflects the error committed after each training iteration. There exist several methods to find an acceptable combination of hyper parameters in the infinite search space, these methods include grid-search, random-search, bayesian hyper parameter optimization, as well as the use of genetic algorithms.

Several libraries and frameworks are available for hyper parameter tuning and model building. The R caret [21] package offers a suite of algorithms for classification and regression. In Python, the scikit-learn [22] library is widely used as it is built on NumPy [23], SciPy [24] and matplotlib [25]. Scikit-learn offers algorithms for supervised and unsupervised learning as well as for hyper parameter tuning and preprocessing tasks. Keras [26] is a high-level open source library that allows implementing models using neural networks. It is implemented in Python but it also offers an interface for its usage with R [27] programming language.

For classification evaluation mostly used metrics are accuracy, precision, recall (sensitivity), specificity and F1-score. The F1-score is preferred when facing class unbalance “since accuracy can be trivially maximized by always predicting the majority class" [19].

After surveying the above methods, the following conclusion can be drawn: protein functional annotation is an active discipline that requires powerful automated or semi-automated systems able to recognise the potential function of sequences even when the query sequence is highly divergent from sequences used to train those systems. In this project a supervised learning-based system is developed in order to identify sequences annotated as GR in non-insect arthropods but with functional shifts like those that occurred in OR of insects.

4 Methodology

This section describes the method used to develop a classifier for OR/GR protein function prediction. The method consists of four main steps: (1) Data retrieval and preparation; (2) Hyper parameter tuning and model training; (3) Model evaluation; (4) Prediction on non-insect arthropods sequences currently annotated as GRs. All the workflow was implemented in Python 3 in this Jupyter Notebook hosted on Google Colaboratory.

Data retrieval

Sequences were obtained from the UniProtKB website querying TrEMBL and Swiss-Prot databases in October 2021. For insect GR sequences, the term “gustatory receptor” is queried. In the advanced filter, insecta was used for taxonomy, and the PFAM domain (pf08395) and the interPro signature (ipr013604) of the insect chemoreceptor protein family were used for filtering. A total of 13,893 sequences were obtained from TrEMBL (non-curated) and 68 from swiss-Prot (curated). In the case of insect OR sequences “odorant receptor” term was queried and pf02949 and ipr004117 were used for filtering. 5,456 non-curated and 54 curated sequences were obtained. The retrieved sequences were downloaded and stored in four files in FASTA format (two for each category, curated and non-curated) for further processing.

Sequence scanning and filtering

To ensure that the downloaded sequences are trustable insect odorant and gustatory receptors, the sequences were scanned using the tool hmmscan from EMBL-EBI API. This tool scans protein sequences against profile-HMM containing databases. The sequences identified in this step were stored and the following filtering criteria could be applied:

1. All hits resulting from the profile-HMM search must include exclusively the pf08395 (for GR) and pf02949 (for OR) domains. Sequences showing one or more hits to a different protein family were discarded.
2. Only hits with an e-value < 1e-6 were retained, i.e., the probability of obtaining a positive result just by chance is very low.
3. Only sequences that show 60% or more sequence aligned with the corresponding domain were considered.

Sequence encoding

In order to encode sequences and feed the model, a standard length needs to be determined. For insect OR and GR sequences, the average length was 400 amino acids. As 98% of the total number of sequences had a length less or equal than 600 amino acids, 600 was taken as the standard length.

Three different encoding methods were used: 1-hot encoding [12], physicochemical properties of amino acids [13] and 10-factor encoding [14] based on a statistical analysis of amino acids physical properties.

For 1-hot encoding every amino acid is replaced by a vector of length 20, for physicochemical encoding a vector of length 6 is used representing the normalized values of 6 physicochemical properties which are hydrophobicity, hydrophilicity, polarity, polarizability, solvation free energy and residue surface area in tripeptide [13]. In the case of 10-factor encoding, a vector of length 10 is used where the first 4 positions correspond to properties related to bend-structure, bulk, beta-structure and hydrophobicity, and the last six positions are mixtures of several physical properties. Thus, encoded sequences have a maximum length of 12,000 (20*600) for 1-hot, 3,600 (6*600) for physicochemical properties and 6,000 (10*600) for 10-factor encoding.

Once sequences were encoded, they were zero-end padded so that all end up having the same length.

Data split

After encoding, all the sequences containing amino acids other than the 20 common proteinogenic were discarded, obtaining a final data set of 15,222 sequences. The OR sequences represented 70.69% of this data set (10,709 non-curated and 52 curated), while the remaining 29.31% corresponded to GR sequences (4,412 non-curated and 49 curated).

Curated sequences were kept apart as an extra subset for evaluating the model. Data partition was done so that 60% of the non-curated sequences were used for training the model, 20% for evaluating and the remaining 20% for testing. Data were shuffled and the same random seed was used to ensure that the same partition is done across all encoding methods.

Dimensionality reduction

Given the way the data were encoded, sparse matrices were obtained and this may increase model complexity, as many variables are used to represent the data (high dimensional data).

Truncated Singular Value Decomposition (SVD) is an effective dimensionality reduction technique when dealing with high sparsity matrices [28] as it doesn’t center the data before computing the singular value decomposition. SVD transformer was fitted with the training dataset and the resulting fitted transformer was used to transform the validation and test sets (they were not used for fitting as they are supposed to be unknown data).

For each type of encoding, truncated SVD was applied such that the data variance explained by the calculated components represent at least 95 percent of the total data variance.

Hyper parameter search and model training

A Keras sequential neural network was used as a classifier. The network was compound by an input layer (layer_0) where the number of units depends on the number of components chosen for each encoding, one dense layer where number of units is a hyper parameter (layer_1), another dense layer of two units (layer_2) that was later used to graphically see how sequences sparse over 2 dimensions and a layer with 1 node (layer_3) with sigmoid activation function as an output layer for binary classification.

To minimize the overfitting of the model a Dropout layer was implemented between layer_1 and layer_2 and a BatchNormalisation layer between layer_2 and the output layer (layer_3). As well, an early stopping was specified for the Area Under the Curve (AUC) for the validation dataset. If after three training epochs a better result in the metric is not obtained then the training loop breaks.

Hyper parameters to be tuned were: number of units and activation function for layer_1, dropout rate, activation function for layer_2 and the network optimizer. As the number of units was also a hyper parameter, the number of trainable parameters may differ among encoding methods.

A Bayesian search of hyper parameters was done using Keras Tuner tool [29], a maximum of ten trials was given using the validation AUC as the objective metric to be maximized (validation dataset was used for comparing different hyper parameter setups).

After the ten trials, the best combination of hyper parameters found was taken and used to retrain the network with those hyper parameters, binary cross entropy was used as the loss function (appropriate for binary classification).

Model evaluation

Test dataset was used to give an unbiased estimate of the performance of the trained model. The following metrics were measured: Accuracy, precision, recall, AUC and F1 score. Confusion matrix, learning curves and Receiving Operating Characteristics (ROC) curves were also obtained for each encoding.

The accuracy metric measures the total number of correct predictions over the total number of predictions, it takes values from 0 to 1 where 1 means that all predictions (all sequences to be classified) are correct and 0 means that all failed.

Precision represents the proportion of sequences classified by the model as the positive class (in this case OR) that are positive. 0 means that any of the sequences classified as OR is OR (all of them would be GR), 1 means that all sequences that the model predicted as OR are actually OR.

Recall or sensitivity measures how many positive cases were detected out of all positive cases. In this case, over the total number of OR sequences how many of them have been classified by the model as OR. 1 means all OR sequences were detected by the model, 0 means that any of the OR sequences was detected.

AUC also varies from 0 to 1 and it measures the classifier skill of distinguishing between classes, the higher the value is the better the classifier is at predicting positive classes as positive and negative classes as negative.

F1 score is the harmonic mean of precision (true positives over all predicted positives) and recall (true positives over all real positives) in this case the positive class (the one labeled as 1) is OR. F1 varies from 0 to 1, one means a perfect precision and recall.

In order to quantify the uncertainty of estimates bootstrap resampling with replacement was done over the test set (3,025 items). One thousand sets of size 3,025 were generated and used to evaluate each classifier, thus metrics could be given with 95% confidence intervals.

Non-insect arthropods sequences

Once classification models were generated using insect OR-GR proteins, they were applied on sequences from non-insect arthropods currently annotated as GRs based on sequence similarity-based searches. In this step, 3,964 non-insect arthropod chemoreceptors provided by the Sánchez-Gracia lab (this dataset includes sequences identified and classified by researchers of this lab, as well as publicly available sequences) are used to obtain the classifiers' predictions.

Saved models as well as SVD objects were loaded to build a pipeline consisting of the following steps:

1. Sequence filtering by total protein length.
2. Sequence encoding.
3. Dimensionality reduction.
4. Prediction of three models.

Predictions were stored in a table where values are probabilities of belonging to the "OR class" reported by classifiers. Each row represents the predictions for a particular sequence and each column the predictions of one of the classifiers for each of the sequences. Probability thresholds were applied to classifiers' predictions so robust and unbiased predictions were obtained.

5 Results

Data retrieval

A total of 19,474 sequences were obtained from UniProt. The most represented class was insect ORs, as it can be seen in Figure 2. Number of sequences by category as well as its percentage over all sequences is shown in Table 2.

Figure 2: Number of sequences by category

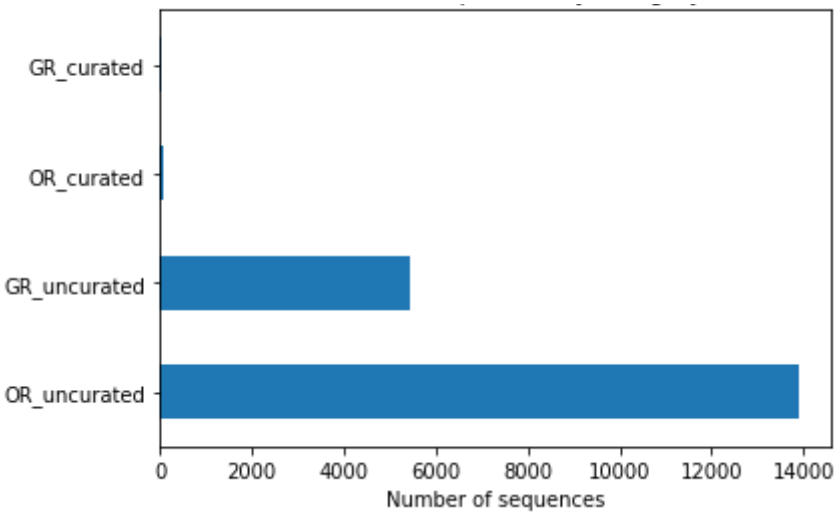


Table 2: Number of sequences and percentage before filtering

Category	Total number	Percentage of all dataset (%)
OR non-curated	13,896	71.35
GR non-curated	5,456	28.01
OR curated	68	0.34
GR curated	54	0.27

Sequences length

After filtering sequences by domain and e-value, a standard length had to be determined before encoding. Initial distribution of filtered sequences' length can be seen in Figures 3 and 4.

Figure 3: Length by category

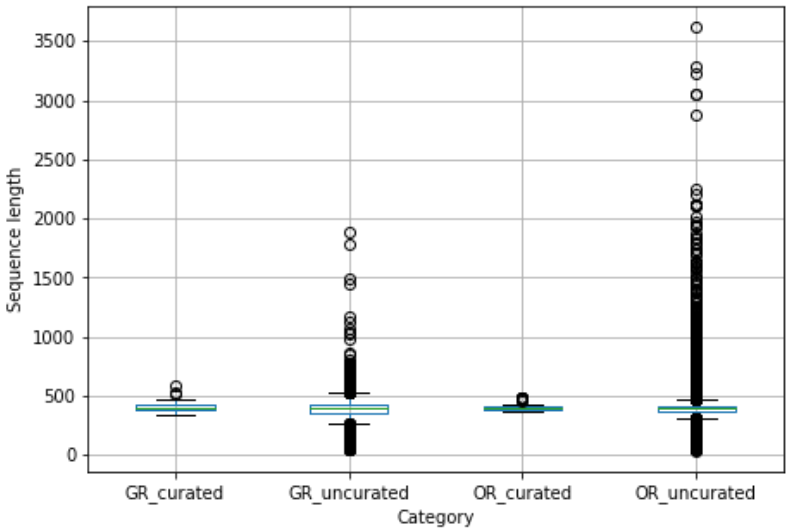
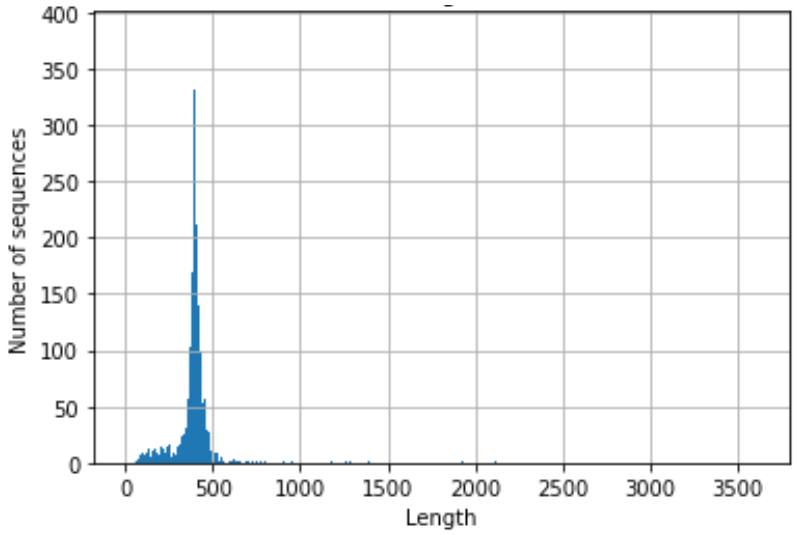


Figure 4: Number of sequences by length



As 98.05% of the total number of sequences showed a length equal to or less than 600 amino acids, length 600 was selected as the standard length. The resulting length distribution after discarding sequences with more than 600 amino acids is shown in Figures 5 and 6.

Figure 5: Length by category (≤600 AA)

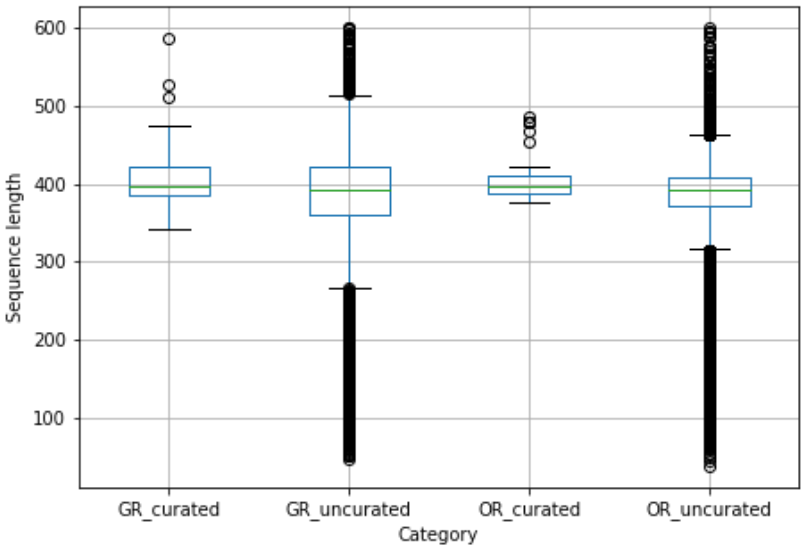
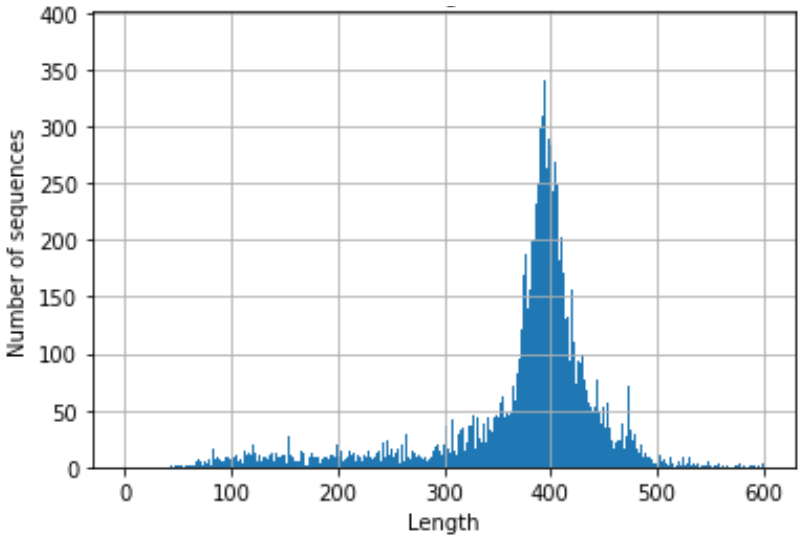


Figure 6: Number of sequences by length (≤600 AA)



Data split

Once sequences have been filtered and sequences containing amino acids other than the 20 common proteinogenic were discarded, the final dataset was composed of a total of 15,222 sequences. Number of sequences by category and proportion can be seen in Table 3.

Table 3: Number of sequences and percentage after filtering

Category	Total number	Percentage of all dataset (%)
OR non-curated	10,709	70.35
GR non-curated	4,412	28.98
OR curated	52	0.34
GR curated	49	0.32

Curated sequences were kept apart as an independent set to test classifiers. The remaining sequences were split such that 60% of the non-curated sequences were used in the training set and the remaining 40% were used for validation and test sets. Number of sequences by subset is shown in Table 4.

Table 4: Number of sequences by subset

Data set	OR	GR	Total
Train	6,423	2,649	9,072
Validation	2,138	886	3,024
Test	2,148	877	3,025
Curated	52	49	101

Dimensionality reduction

Truncated SVD was applied to reduce data dimensionality. For each encoding method a few components had to be chosen in a way that at least 95% of the variance present in the data was explained by the number of components. Explained variance with respect to the number of components is shown in Figures 7-9.

Figure 7: Cumulative explained variance (1-hot)

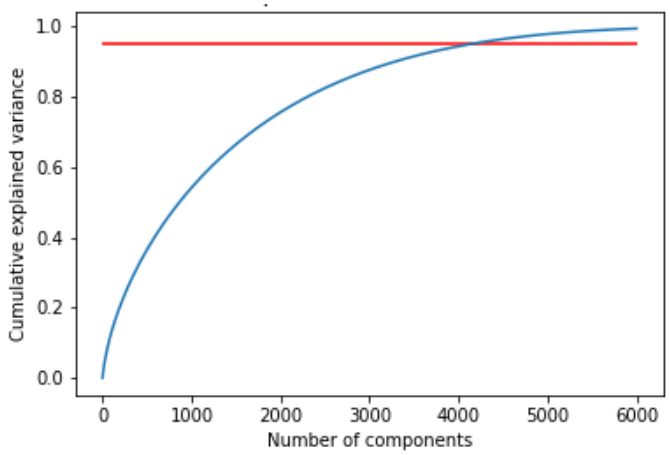


Figure 8: Cumulative explained variance (10-factor)

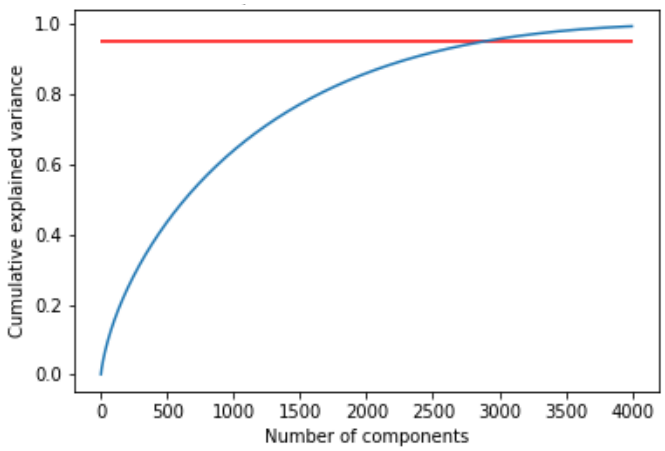
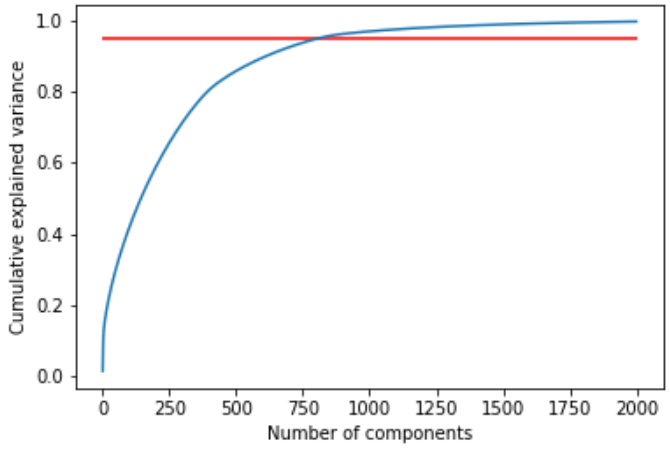


Figure 9: Cumulative explained variance (phy-chem)



For 1-hot encoding the number of chosen components was 4,250 keeping a 95.35% of the variance, this represents a decrease of 64.59% in the number of variables used to represent sequences. In the case of 10-factor 3,000 components explained 95.83% of the variance (50% decrease) and for physicochemical encoding 850 components explained 95.66% (76.39% decrease as originally 3,600 variables were used).

Hyper parameter search and model training

The general architecture of the neural network can be seen in Figure 10. Found hyper parameters for each encoding is shown in Table 5.

Figure 10: Neural network architecture

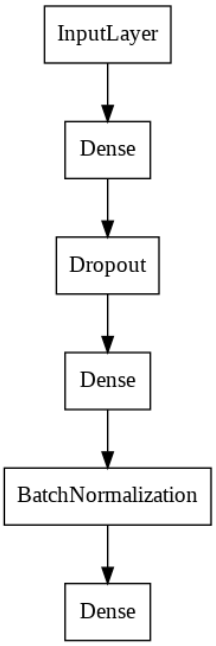


Table 5: Neural networks hyper parameters

Encoding	Network optimizer	Dropout rate	Dense layer_1 hidden units	Dense layer_2 activation function
1-hot	Adam	0.2	30	tanh
10-factor	SGD	0.5	300	ReLU
phy-chem	Adam	0.2	30	tanh

Figures 11-13 show the learning curves for each encoding method. It can be seen that for all encoding methods a high accuracy rate is achieved at the first training epochs. However, some overfitting is present as training accuracy is higher than validation accuracy in the last training epochs.

Figure 11: Learning curves (1-hot)

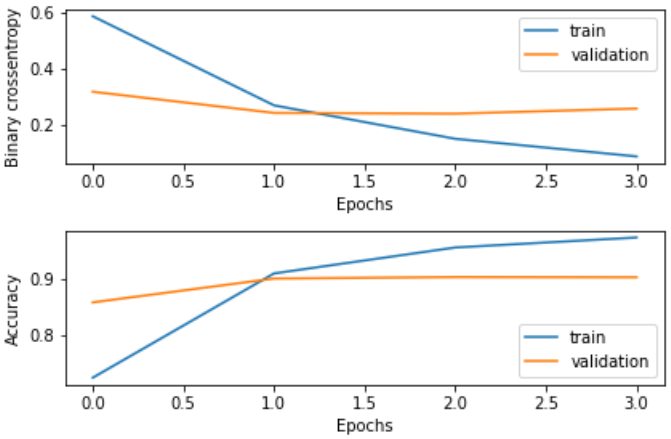


Figure 12: Learning curves (10-factor)

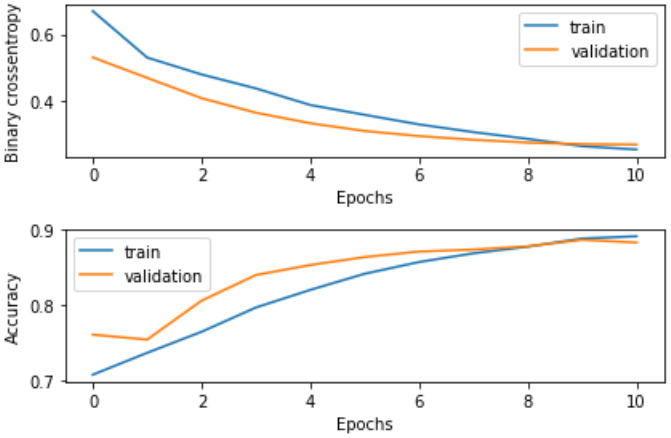
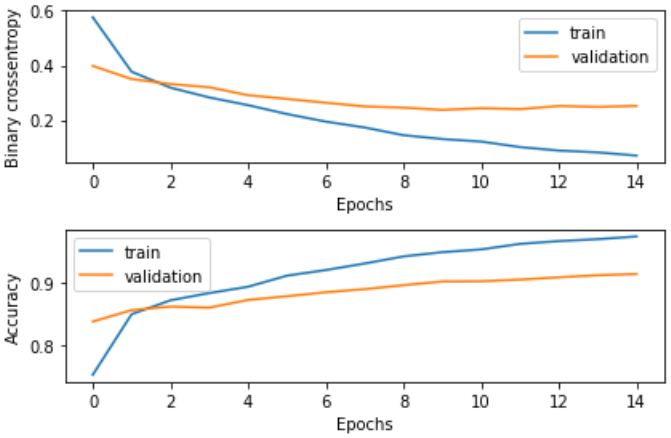


Figure 13: Learning curves (phy-chem)



The second dense layer that contains hidden units (neurons) was used as a latent space in order to visualize the embedded sequences in a two-dimensional space. For all encoding methods the network makes a distinction between OR and GR sequences as it can be seen in Figures 14-16. Axis values differ between encoding methods as for 1-hot encoding and physicochemical encoding the activation function found was the hyperbolic tangent (tanh, Table 2) whose domain is a real number between -1 and 1. For the 10-factor encoding, the activation function for the second dense layer was the Rectified Linear Unit (ReLU, Table 2) whose domain is ≥0.

Figure 14: Embedded sequences (1-hot)

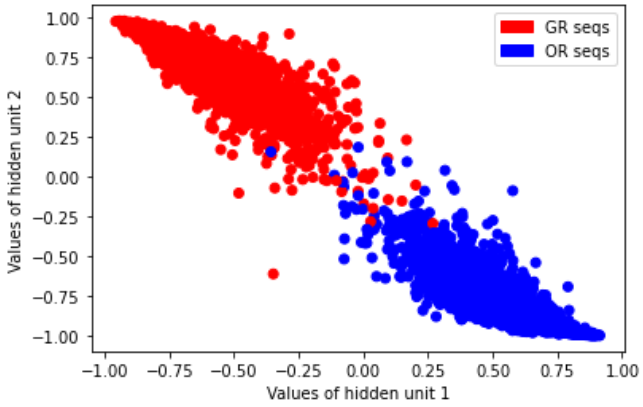


Figure 15: Embedded sequences (10-factor)

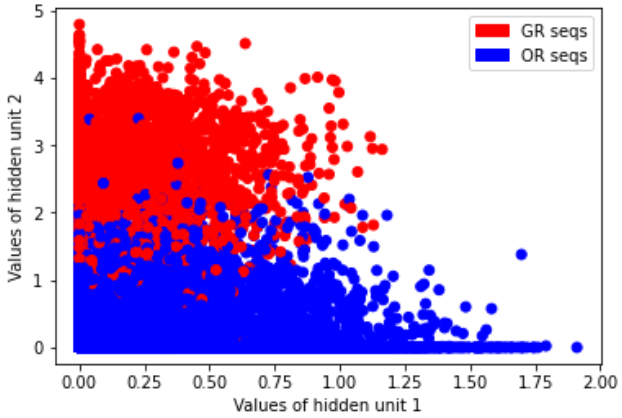
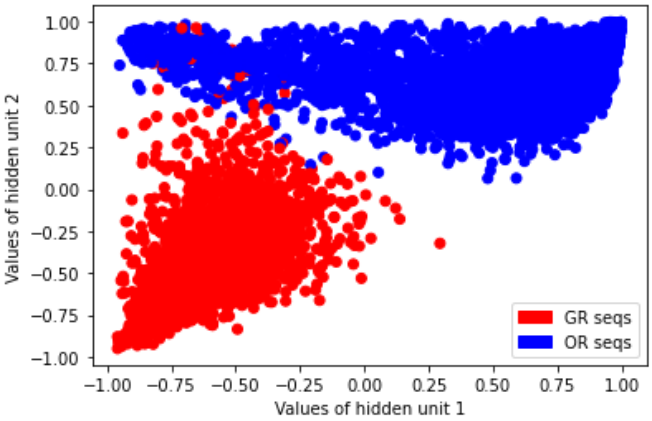


Figure 16: Embedded sequences (phy-chem)



Model evaluation

For each encoding method, the ROC curve and the confusion matrix were obtained using the test set, overall performance is quite alike between methods:

Figure 17: ROC curve (1-hot)

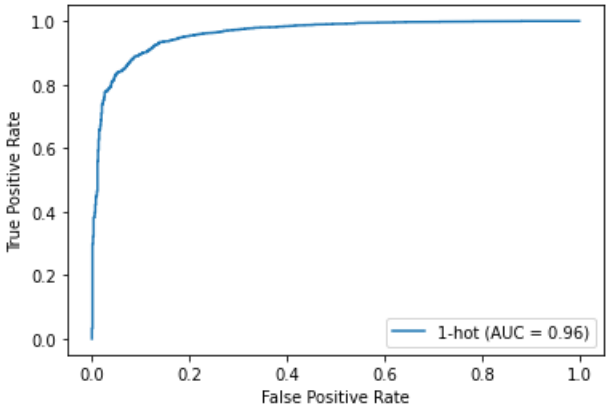


Figure 18: Confusion matrix (1-hot)

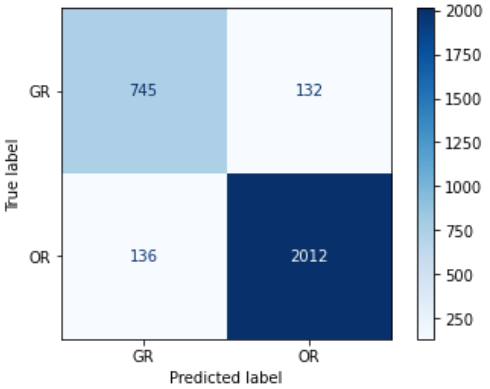


Figure 19: ROC curve (10-factor)

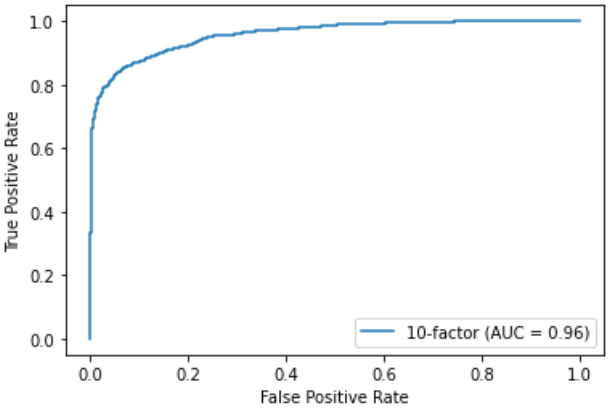


Figure 20: Confusion matrix (10-factor)

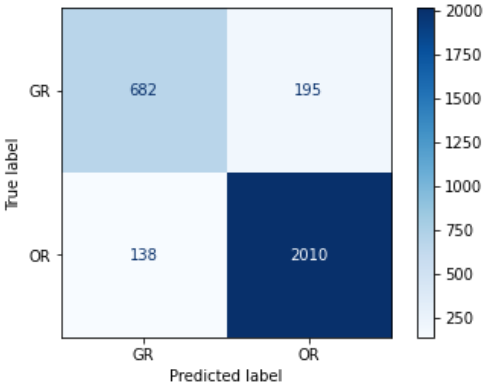


Figure 21: ROC curve (phy-chem)

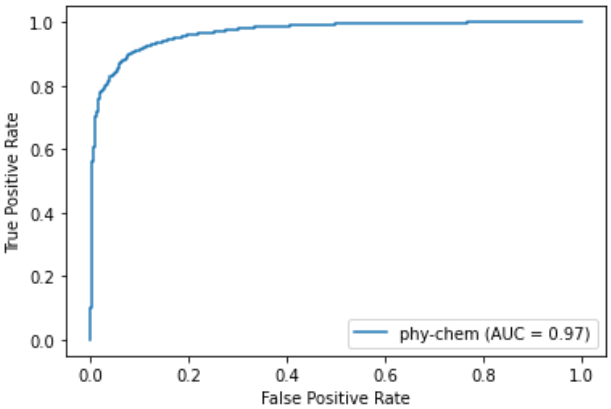
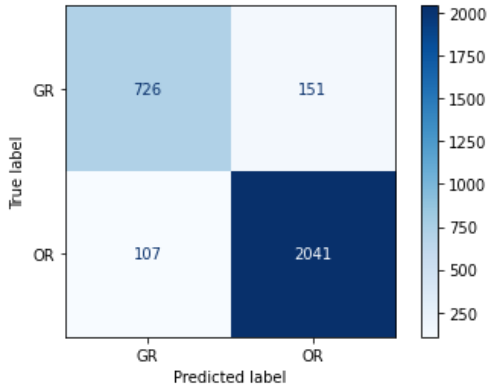


Figure 22: Confusion matrix (phy-chem)



Metrics for each encoding method and dataset were calculated:

Table 6: Metrics for each encoding method

Encoding	Data set	Accuracy	AUC	Precision	F1	Recall
1-hot	Train	0.997	0.999	0.998	0.998	0.998
	Validation	0.904	0.956	0.930	0.932	0.933
	Test	0.911	0.961	0.938	0.937	0.936
	Curated	1	1	1	1	1
10-factor	Train	0.952	0.990	0.956	0.966	0.977
	Validation	0.883	0.946	0.906	0.918	0.931
	Test	0.889	0.958	0.911	0.923	0.935
	Curated	0.980	1	0.962	0.981	1
phy-chem	Train	0.995	0.999	0.993	0.996	0.999
	Validation	0.914	0.961	0.927	0.940	0.953
	Test	0.914	0.966	0.931	0.940	0.950
	Curated	0.990	1	0.981	0.990	1

1000 bootstrap samples with replacement of size 3,025 were generated using the test set, so metrics could be calculated with 95% confidence interval:

Table 7: Metrics confidence intervals

Encoding	Accuracy	AUC	Precision	F1	Recall
1-hot	0.911 (0.901-0.920)	0.893 (0.880-0.904)	0.938 (0.927-0.947)	0.937 (0.930-0.944)	0.936 (0.926-0.946)
10-factor	0.889 (0.878-0.901)	0.856 (0.842-0.870)	0.911 (0.899-0.923)	0.923 (0.915-0.932)	0.935 (0.925-0.945)
phy-chem	0.914 (0.905-0.924)	0.889 (0.876-0.901)	0.931 (0.920-0.941)	0.940 (0.933-0.947)	0.950 (0.941-0.959)
Average	**0.904±0.011**	**0.879±0.016**	**0.926±0.011**	**0.933±0.007**	**0.940±0.006**

Final model evaluation

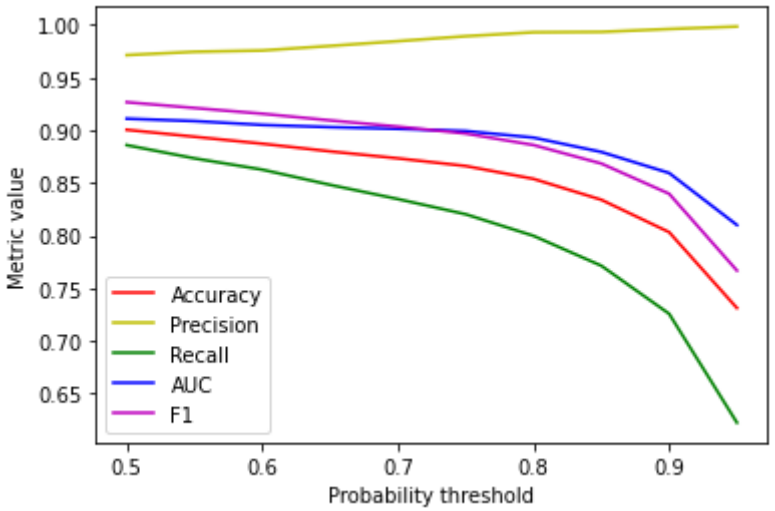
Different thresholds were applied to the classifiers' OR class probabilities. Sequences for which the three developed classifiers showed a probability equal to or higher than the specified threshold were identified as the OR category (i.e. candidates to have undergone a functional shift like that of insect ORs from a GR). In Table 8 metrics obtained for each probability threshold are shown:

Table 8: Metrics for probability thresholds

Threshold	Accuracy	AUC	Precision	F1	Recall
0.50	**0.900**	**0.911**	**0.971**	**0.926**	**0.885**
0.55	0.893	0.908	0.974	0.921	0.873
0.60	0.887	0.905	0.975	0.915	0.862
0.65	0.880	0.903	0.980	0.909	0.848
0.70	0.873	0.901	0.984	0.903	0.834
0.75	0.866	0.899	0.989	0.896	0.820
0.80	0.853	0.893	0.993	0.886	0.799
0.85	0.834	0.879	0.993	0.868	0.771
0.90	0.803	0.859	0.996	0.839	0.725
0.95	0.731	0.810	0.998	0.766	0.622

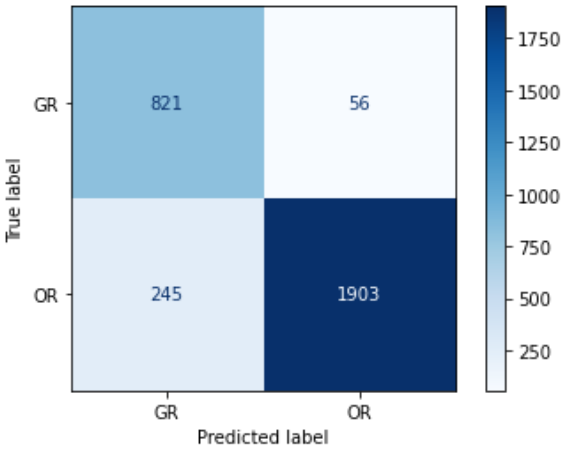
Figure 23 shows how the metrics values change respect to the different probability thresholds. After a threshold probability of 0.80, the precision is nearly 100%.

Figure 23: Line plot of metrics



The confusion matrix obtained for the final model applying 0.5 as the probability threshold is shown in Figure 24:

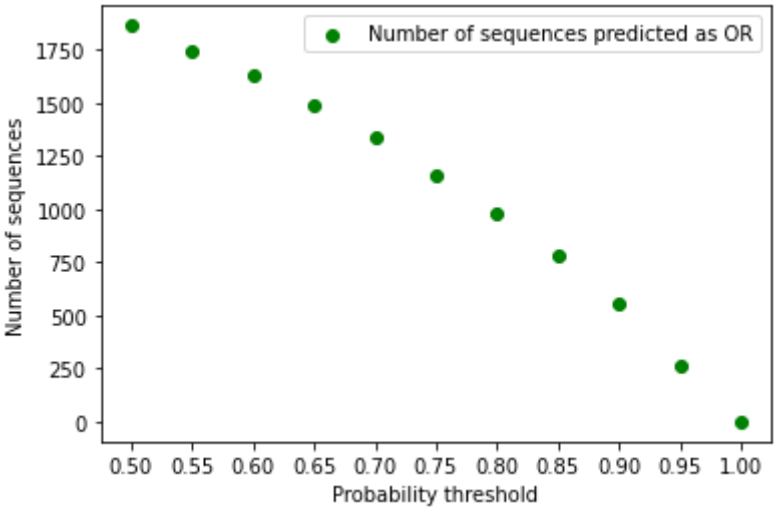
Figure 24: Confusion matrix (final model)



Non-insect arthropod sequences

From a total of 3,964 sequences, 3,956 had a length equal or less to 600 amino acids. Figure 24 shows the number of predicted OR sequences for different probability thresholds.

Figure 25: Number of OR sequences by probability threshold



For each classifier prediction probabilities are shown in Figures 25-27:

Figure 26: Prediction probabilities (1-hot)

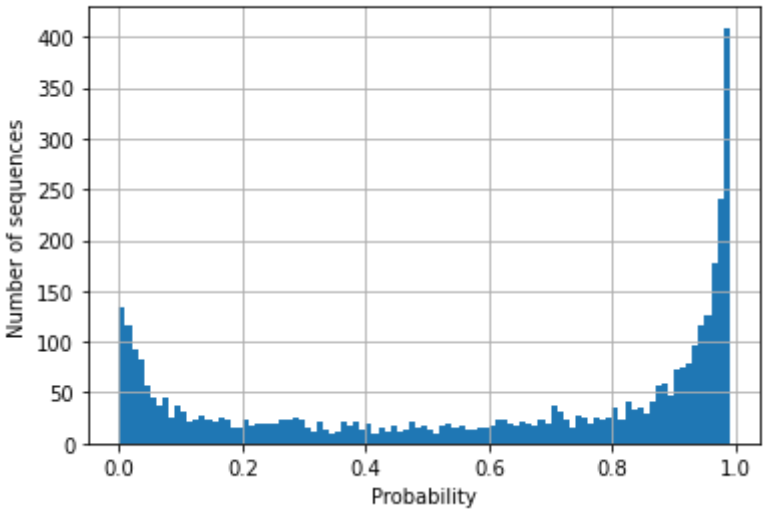


Figure 27: Prediction probabilities (10-factor)

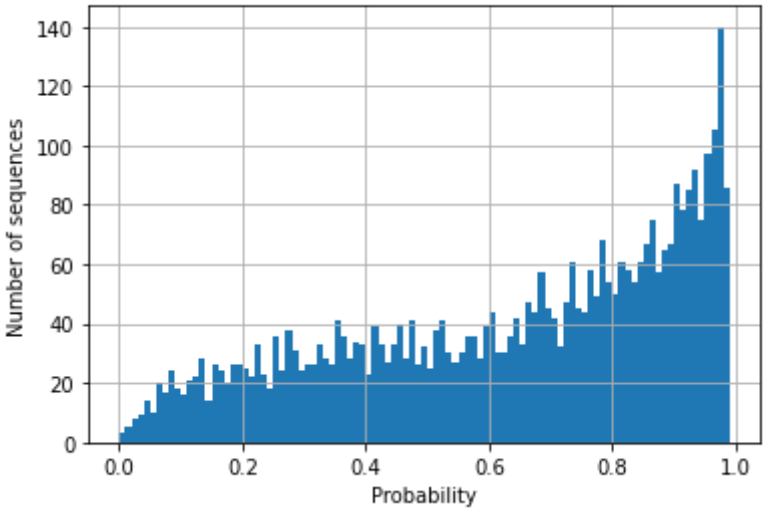
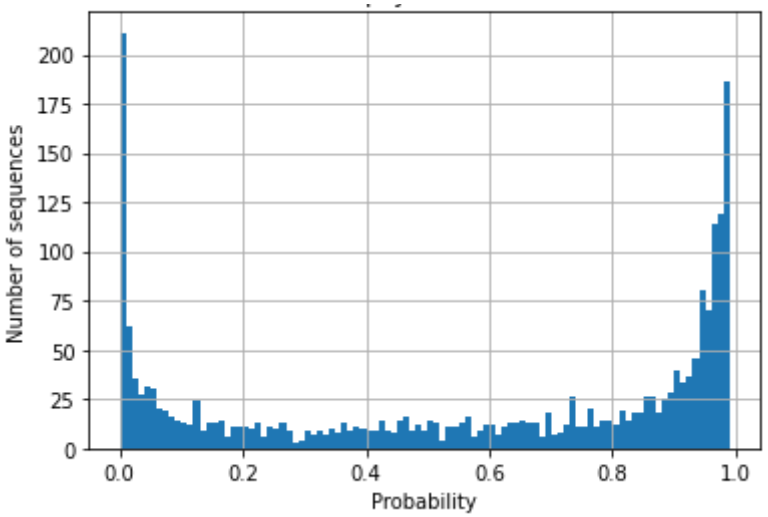


Figure 28: Prediction probabilities (phy-chem)



A total of 265 sequences were predicted to belong to the OR category (i.e. identifying a functional shift like that of insect ORs from a GR) with a probability higher than 95% by the three classifiers (see the annex).

6 Discussion

The results obtained will be discussed, from the dataset distribution to each classifier performance, as well as the final model performance.

The dataset were imbalanced given that OR was the most represented class (over 70% of all retrieved sequences) and this may lead to a poor performance of the classifiers given that the data are biased. However the positive class in this project was OR and keeping all OR sequences was relevant so that the classifiers could be trained taking into account all the variability present in sequences of the OR class. A balanced version of the dataset was obtained via downsampling the majority class but yet no improvement in performance metrics was observed.

As expected, curated sequences were scarce, representing less than 1% of the total number of retrieved sequences. Even though the results obtained in the performance metrics after testing the classifiers were in line with the results obtained from the validation and test sets, they are not as representative as others due to the sample size (just 101 curated sequences versus 3,025 sequences in the test set).

The difference in the accuracy metric in Table 6 between training and validation sets was 7-9%. The present overfitting was unavoidable despite including a dropout layer, batch normalisation layer in the neural network. However, thanks to an early stopping call-back, it was possible to minimise the overfitting and avoid a higher generalization error.

The stochastic nature of the algorithm is visible in Figures 14-16, as embedded sequences group in different ways for different encoding methods. Even for the same encoding method, a different plot would be obtained if the network is retrained. Sequences form two main clusters (one for GR class and another for the OR class). In 1-hot and physicochemical encoding plots, a clear separation between the two clusters is appreciated, this is not the case for 10-factor encoding.

After bootstrapping over the test set to obtain metrics with 95% confidence intervals, average AUC reported that there is a 87.9% chance (AUC=0.879±0.016) that the three classifiers will be able to distinguish between positive class and negative class, namely OR and GR category, respectively.

On average, the percentage of OR sequences identified by the classifiers was 94% (Recall=0.940±0.006) and the percentage of sequences predicted as OR that were actual OR sequences was 92.6% (Precision=0.926±0.011).

Given the tradeoff between recall and precision, the F1 score was calculated as it combines both metrics and using one metric allows to compare different models. The F1 score was quite the same for the three classifiers, on average 0.933. Apparently, there is no advantage in using one encoding method over another.

Considering the performance of the final model (the three developed classifiers combined), the AUC and the precision obtained applying a 0.5 probability threshold (see Table 8) outperformed the average metrics. When the three classifiers are assembled, predictions are more robust and unbiased. A 97.10% of precision was obtained, meaning a 4.5% boost with respect to the average precision of the three classifiers. The precision value increases as the probability threshold increases. The opposite happens with the other calculated metrics as it can be seen in Table 8 and Figure 23. This property of the model was relevant for this project given that a highly precise model could identify potential receptors for volatile hydrophobic molecules derived from GRs in non-insect arthropod lineages.

When applying the model to non-insect arthropod sequences, a clear descent tendency was observed in the number of sequences classified as OR as the probability threshold increased (see figure 24). Anyway, 265 sequences annotated as GR by sequence similarity-based methods were classified as OR by the model developed with at least a 95% probability given by the three classifiers, this represent a 6.69% of the total number of classified sequences (3,956). These results open the door to further research on the candidate sequences in order to analyse a possible evolutionary convergence, since these non-insect arthropods may have developed proteins with functional properties similar to those of OR in insects.

Respect to each classifier probability distributions, it can be appreciated a skewness to 1 (specially for 1-hot and 10-factor classifiers), as there was a high number of sequences for which the given probability of belonging to the OR category was nearly 1. This effect could be attributed to the class imbalance present in the dataset. However, the present class imbalance in the dataset did not negatively affect the AUC and F1 metrics, which are useful when dealing with imbalanced binary classification problems.

7 Conclusions

7.1 Conclusions

Results reported by the three developed classifiers, as well as by the final classifier, prove that it is possible to classify protein sequences relying solely on its amino acids data.

No relevant differences were found between encoding methods.

When predicting functionality of non-insect arthropod protein sequences labeled as GR by similarity based systems, a list of sequences were identified as candidates that potentially have similar functional properties to those of insect ORs.

More evidence regarding functional properties of candidate sequences needs to be collected in order to check the validity of the classification of candidate sequences.

The first main objective of this project was achieved since a supervised learning based system was developed in order to classify insect sequences between odorant and gustatory receptors. As well, the code and data used to develop the model is publicly available on Google colab. However a Python package creation was not possible due to time constraints and that special effort was taken in achieving the first objective and making the code clear and intuitive for other users.

7.2 Future work

Given time constraints, non-insect arthropod sequences classified as OR by the classifier could not be further analysed. A phylogenetic tree could be generated with candidate OR sequences to see if they are phylogenetically close. As well, an expression analysis could be performed in order to check if candidate proteins are present in the same tissue of the organism and in which organs they are present.

From the perspective of the model, in order to test if an improvement is achieved in the model performance:

A. More variables containing information about the 3-D structure of proteins could be included.
B. Additional hyper parameters as the learning rate of the network could be included to be tuned.
C. Another architecture can be implemented as other models have been pre trained with Pfam protein sequences [30] and its structure and parameters can be adapted and retrained with insect OR/GR sequences, this is known as transfer learning.

To analyse the effect of class imbalance in the distribution of probabilities given by the three classifiers, different techniques could be tested upsampling the minority class (although this may lead to a poor model generalization of the GR class) or data augmentation.

7.3 Planning monitoring

Scheduled tasks were accomplished on time, however extra time was needed for project documentation. No changes had to be applied to ensure project success.

Respect to methodology, as initially planned supervised learning was applied, as “black box” algorithms have been used to develop the classifiers so technically it is a deep learning implementation.

8 Glossary

API: Application Programming Interface.

ARBA: Association Rule Based Annotator. Automated system used to annotate sequences in the UniProtKB TrEMBL database. Rule mining techniques are used to generate concise annotation models with the highest representativeness and coverage for annotation, based on the properties of InterPro group membership and taxonomy.

AUC: Area Under the ROC Curve.

BLOSUM: BLOcks of amino acid SUbstitution Matrix. Amino acid substitution matrix used to align evolutionary divergent protein sequences. It is based on local alignments.

DL: Deep Learning. Branch of ML where algorithms used are based on Artificial Neural Networks (ANN). It is considered Deep as many hidden layers with multiple units (neurons) can be used when implementing the algorithm.

EMBL-EBI: European Molecular Biology Laboratory - European Bioinformatics Institute.

GO: Gene Ontology. The Gene Ontology knowledgebase is the world's largest source of information on the functions of genes. This knowledge is both human-readable and machine-readable, and is a foundation for computational analysis of large-scale molecular biology and genetics experiments in biomedical research.

GRs: Gustatory receptors. Proteins implied in the sensation of taste as it binds to flavour compounds in food and other elements when they get in contact.

HMM: Hidden Markov Model.

Hyper parameters: All possible setups for an algorithm. The chosen setup will govern how parameters of the algorithm are estimated as it has a high impact over the learning process.

IRs: Ionotropic receptors. Membrane-bound receptor proteins that respond to ligand binding by opening an ion channel and allowing ions to flow into the cell, either increasing or decreasing the likelihood that an action potential will fire.

iGluR: Ionotropic glutamate receptors.

ML: Machine Learning. Discipline under the field of Artificial Intelligence, whose aim is to create a software able to make predictions (as classification or regression tasks) based on an algorithm and historical data.

Model: Internal representation of the pattern to be recognised by an algorithm. Once the training process of an algorithm is completed, the estimated parameters are used to make predictions.

OBPs: Odorant binding proteins. Soluble proteins secreted by auxiliary cells surrounding odorant receptor neurons (ORNs), it is believed that they act as odorant transporters, delivering the odorant molecules to olfactory receptors in the cell membrane of sensory neurons.

ORs: Odorant receptors. They are chemoreceptors expressed in the cell membranes of olfactory receptor neurons and are responsible for the detection of odorants which give rise to the sense of smell.

ORN: Odorant receptor neuron is a sensory neuron within the olfactory system.

PAM: Point Accepted Mutation matrices are amino acid substitution matrices that encode the expected evolutionary change at the amino acid level. Each PAM matrix is designed to compare two sequences which are a specific number of PAM units apart.

PFAM: Protein family.

ReLU: Rectified Linear Unit.

ROC: Receiver Operating Characteristics.

SGD: Stochastic Gradient Descent.

SVD: Singular Value Decomposition.

Swiss-Prot: Protein database in UniProtKB where records are manually annotated and reviewed (curated by experts).

TrEMBL: Protein database in UniProtKB that contains all the translations of European Molecular Biology Laboratory (EMBL) nucleotide sequence database. Records are automatically annotated.

UniProtKB: UniProt Knowledgebase. Freely accessible repository of protein's data where most records are provided by genome sequencing projects.

9 References

[1] Yan, H., Jafari, S., Pask, G., Zhou, X., Reinberg, D., Desplan, C., 2020. Evolution, developmental expression and function of odorant receptors in insects. Journal of Experimental Biology 223. doi:10.1242/jeb.208215

[2] Robertson, H.M., 2019. Molecular Evolution of the Major Arthropod Chemoreceptor Gene Families. Annual Review of Entomology 64, 227–242. doi:10.1146/annurev-ento-020117-043322

[3] Orlando, G., Raimondi, D., Khan, T., Lenaerts, T., Vranken, W.F., 2017. SVM-dependent pairwise HMM: an application to protein pairwise alignments. Bioinformatics 33, 3902–3908. doi:10.1093/bioinformatics/btx391

[4] The UniProt Consortium, 2020. UniProt: the universal protein knowledgebase in 2021. Nucleic Acids Research 49, D480–D489. doi:10.1093/nar/gkaa1100

[5] Van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. Scotts Valley, CA: CreateSpace.

[6] github. (2021). GitHub. Retrieved from https://github.com/

[7] Sánchez-Gracia, A., Vieira, F.G., Almeida, F.C., Rozas, J., 2011. Comparative Genomics of the Major Chemosensory Gene Families in Arthropods. eLS. doi:10.1002/9780470015902.a0022848

[8] Wicher, D., Miazzi, F., 2021. Functional properties of insect olfactory receptors: ionotropic receptors and odorant receptors. Cell and Tissue Research 383, 7–19. doi:10.1007/s00441-020-03363-x

[9] Pask, G.M., Slone, J.D., Millar, J.G., Das, P., Moreira, J.A., Zhou, X., Bello, J., Berger, S.L., Bonasio, R., Desplan, C., Reinberg, D., Liebig, J., Zwiebel, L.J., Ray, A., 2017. Specialized odorant receptors in social insects that detect cuticular hydrocarbon cues and candidate pheromones. Nature Communications 8. doi:10.1038/s41467-017-00099-1

[10] Blum, M., Chang, H.-Y., Chuguransky, S., Grego, T., Kandasaamy, S., Mitchell, A., Nuka, G., Paysan-Lafosse, T., Qureshi, M., Raj, S., Richardson, L., Salazar, G.A., Williams, L., Bork, P., Bridge, A., Gough, J., Haft, D.H., Letunic, I., Marchler-Bauer, A., Mi, H., Natale, D.A., Necci, M., Orengo, C.A., Pandurangan, A.P., Rivoire, C., Sigrist, C.J.A., Sillitoe, I., Thanki, N., Thomas, P.D., Tosatto, S.C.E., Wu, C.H., Bateman, A., Finn, R.D., 2020. The InterPro protein families and domains database: 20 years on. Nucleic Acids Research 49, D344–D354. doi:10.1093/nar/gkaa977

[11] The UniProt Consortium, 2018. UniProt: a worldwide hub of protein knowledge. Nucleic Acids Research 47, D506–D515. doi:10.1093/nar/gky1049

[12] Jing, X., Dong, Q., Hong, D., Lu, R., 2020. Amino Acid Encoding Methods for Protein Sequences: A Comprehensive Review and Assessment. IEEE/ACM Transactions on Computational Biology and Bioinformatics 17, 1918–1931. doi:10.1109/tcbb.2019.2911677

[13] Yousef, A., Charkari, N.M., 2015. A novel method based on physicochemical properties of amino acids and one class classification algorithm for disease gene identification. Journal of Biomedical Informatics 56, 300–306. doi:10.1016/j.jbi.2015.06.018

[14] Kidera, A., Konishi, Y., Oka, M., Ooi, T., Scheraga, H.A., 1985. Statistical analysis of the physical properties of the 20 naturally occurring amino acids. Journal of Protein Chemistry 4, 23–55. doi:doi.org/10.1007/bf01025492

[15] Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S.A.A., Ballard, A.J., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A.W., Kavukcuoglu, K., Kohli, P., Hassabis, D., 2021. Highly accurate protein structure prediction with AlphaFold. Nature 596, 583–589. doi:10.1038/s41586-021-03819-2

[16] Ofer, D., Brandes, N., Linial, M., 2021. The language of proteins: NLP, machine learning & protein sequences. Computational and Structural Biotechnology Journal 19, 1750–1758. doi:10.1016/j.csbj.2021.03.022

[17] Lopez-del Rio, A., Martin, M., Perera-Lluna, A., Saidi, R., 2020. Effect of sequence padding on the performance of deep learning models in archaeal protein functional prediction. Scientific Reports 10. doi:10.1038/s41598-020-71450-8

[18] Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G., 2000. Gene Ontology: tool for the unification of biology. Nature Genetics 25, 25–29. doi:10.1038/75556

[19] Bonetta, R., Valentino, G., 2019. Machine learning techniques for protein function prediction. Proteins: Structure, Function, and Bioinformatics 88, 397–413. doi:10.1002/prot.25832

[20] Vu, T.T.D., Jung, J., 2021. Protein function prediction with gene ontology: from traditional to deep learning models. PeerJ 9, e12019. doi:10.7717/peerj.12019

[21] Kuhn, M., 2008. Building Predictive Models inRUsing thecaretPackage. Journal of Statistical Software 28. doi:10.18637/jss.v028.i05

[22] Pedregosa, F., Varoquaux, Ga"el, Gramfort, A., Michel, V., Thirion, B., Grisel, O., … others. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825–2830.

[23] Harris, C.R., Millman, K.J., van der Walt, S.J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N.J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M.H., Brett, M., Haldane, A., del Río, J.F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., Oliphant, T.E., 2020. Array programming with NumPy. Nature 585, 357–362. doi:10.1038/s41586-020-2649-2

[24] Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Millman, K.J., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C.J., Polat, İ., Feng, Y., Moore, E.W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., Vijaykumar, A., Bardelli, A.P., Rothberg, A., Hilboll, A., Kloeckner, A., Scopatz, A., Lee, A., Rokem, A., Woods, C.N., Fulton, C., Masson, C., Häggström, C., Fitzgerald, C., Nicholson, D.A., Hagen, D.R., Pasechnik, D.V., Olivetti, E., Martin, E., Wieser, E., Silva, F., Lenders, F., Wilhelm, F., Young, G., Price, G.A., Ingold, G.-L., Allen, G.E., Lee, G.R., Audren, H., Probst, I., Dietrich, J.P., Silterra, J., Webber, J.T., Slavič, J., Nothman, J., Buchner, J., Kulick, J., Schönberger, J.L., de Miranda Cardoso, J.V., Reimer, J., Harrington, J., Rodríguez, J.L.C., Nunez-Iglesias, J., Kuczynski, J., Tritz, K., Thoma, M., Newville, M., Kümmerer, M., Bolingbroke, M., Tartre, M., Pak, M., Smith, N.J., Nowaczyk, N., Shebanov, N., Pavlyk, O., Brodtkorb, P.A., Lee, P., McGibbon, R.T., Feldbauer, R., Lewis, S., Tygier, S., Sievert, S., Vigna, S., Peterson, S., More, S., Pudlik, T., Oshima, T., Pingel, T.J., Robitaille, T.P., Spura, T., Jones, T.R., Cera, T., Leslie, T., Zito, T., Krauss, T., Upadhyay, U., Halchenko, Y.O., Vázquez-Baeza, Y., 2020. SciPy 1.0: fundamental algorithms for scientific computing in Python. Nature Methods 17, 261–272. doi:10.1038/s41592-019-0686-2

[25] Hunter, J.D., 2007. Matplotlib: A 2D Graphics Environment. Computing in Science & Engineering 9, 90–95. doi:10.1109/mcse.2007.55

[26] Chollet, F., & others. (2015). Keras. GitHub. Retrieved from https://github.com/fchollet/keras

[27] R Core Team. (2016). R: A Language and Environment for Statistical Computing. Vienna, Austria. Retrieved from https://www.R-project.org/

[28] Nathan Halko, Per-Gunnar Martinsson, Joel A. Tropp (2011). “Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions”, SIAM Rev., Survey and Review section, Vol. 53, num. 2, pp. 217-288

[29] O’Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., & Others. (2019). Keras Tuner. Opgehaal van https://github.com/keras-team/keras-tuner

[30] Bileschi, M. L., Belanger, D., Bryant, D., Sanderson, T., Carter, B., Sculley, D., DePristo, M. A., & Colwell, L. J. (2019). *Using deep learning to annotate the protein universe*. Cold Spring Harbor Laboratory. http://dx.doi.org/10.1101/626507

Annex

Predictions with high probability (i.e. probability ≥ 0.95):

ID	1-hot	phy-chem	10-factor
AgenCR58p	0.97458696	0.9999529	0.9788811
AgenCR63p	0.97074604	0.9999529	0.97450894
AgenCR96p	0.97813356	0.99995273	0.9800757
AgenCR97p	0.9712888	0.9999528	0.9796369
AgenCR107p	0.9830755	0.9999529	0.9806974
AgenCR108p	0.9925222	0.9999529	0.9816097
AgenCR126p	0.9861705	0.9999529	0.9584303
AgenCR130p	0.98886454	0.9999529	0.97812915
AgenCR133p	0.96507424	0.9999528	0.9753275
AgenCR134p	0.97599775	0.9963298	0.953694
AgenCR136p	0.98523164	0.99638855	0.96503097
CexiCR11	0.9835149	0.9893968	0.9716803
CexiCR22	0.9806907	0.997442	0.98004246
CexiCR26	0.97443473	0.99940324	0.965013
CexiCR66	0.99042153	0.976508	0.96350217
CexiCR106	0.99397933	0.9988382	0.9804908
CexiCR118	0.9518727	0.9623221	0.972384
CexiCR139	0.97641206	0.97736204	0.9572655
CexiCR193	0.9752791	0.99936134	0.98158586
CexiCR207	0.9893602	0.98821473	0.9725101
CexiCR212	0.982067	0.9841177	0.9594083
CexiCR222	0.98921347	0.99562466	0.9533597
CexiCR238	0.9839988	0.9997306	0.97838295
CexiCR302	0.9891607	0.9713944	0.9592942
CexiCR317	0.9903275	0.99665546	0.973382

CexiCR326	0.96170115	0.999866	0.9515071
CexiCR331	0.99520767	0.99864167	0.95360315
CexiCR347	0.95995414	0.9951618	0.959925
CexiCR367	0.9871373	0.99816084	0.97838295
CexiCR368	0.96488106	0.9988103	0.9707079
CexiCR373	0.966154	0.99955773	0.9620288
CexiCR401	0.98345125	0.9997623	0.9809502
CexiCR436	0.9877397	0.9996661	0.96655345
CexiCR476	0.9860766	0.9992855	0.97960126
CexiCR478	0.9897023	0.9675642	0.9589053
CexiCR486p	0.98529327	0.9998967	0.9778303
CexiCR494p	0.9820329	0.9999528	0.9787427
CexiCR495p	0.9632057	0.99990755	0.9682011
CexiCR501p	0.9913549	0.9595239	0.9510436
CexiCR510p	0.97445446	0.999953	0.9806144
CexiCR515p	0.98928165	0.9999519	0.97543395
CexiCR533p	0.99286515	0.99995285	0.9784896
CexiCR538p	0.9577372	0.99995285	0.9799752
CexiCR543p	0.99218196	0.9999528	0.963436
CexiCR579p	0.9913297	0.9999529	0.9810655
CexiCR597p	0.963057	0.9986541	0.9587867
CexiCR604p	0.9884124	0.9999524	0.97924626
CexiCR607p	0.98352706	0.99995035	0.9801469
CexiCR627p	0.9949839	0.99968296	0.9670988
CexiCR632p	0.9552773	0.9999516	0.9756615
CexiCR689p	0.98052907	0.9999528	0.9591012
CexiCR694p	0.9949267	0.9999526	0.9795261
CexiCR699p	0.9662219	0.9999521	0.95098877
CexiCR703p	0.98790634	0.9999523	0.9593667
CexiCR715p	0.97651756	0.9999529	0.9784788

CexiCR723p	0.96217865	0.9999528	0.977767
CexiCR724p	0.98651224	0.9999528	0.9528227
CexiCR738p	0.9889263	0.9999435	0.96619356
CexiCR745p	0.98481137	0.99995255	0.9601382
CexiCR747p	0.9881005	0.999953	0.98038065
CexiCR769p	0.97409815	0.9999501	0.9543307
CexiCR783p	0.9896047	0.9999528	0.9792989
CexiCR795p	0.98999727	0.9999528	0.9810554
CexiCR809p	0.98162544	0.9999404	0.9781618
CexiCR813p	0.9531952	0.999953	0.95221514
CexiCR818p	0.9871056	0.99995255	0.97891843
CexiCR821p	0.98379683	0.9999527	0.98012674
CexiCR827p	0.98433286	0.9999527	0.9784107
CexiCR829p	0.97806966	0.9999205	0.97732854
CexiCR838p	0.9604186	0.9999529	0.97838295
CexiCR839p	0.99332535	0.9953648	0.97838295
CexiCR866p	0.9785737	0.99526864	0.9711959
CexiCR869p	0.9862786	0.9999528	0.95109713
CexiCR872p	0.99216944	0.99353826	0.9506701
CexiCR873p	0.98824704	0.99989355	0.9821655
CexiCR879p	0.9920686	0.9999528	0.98057246
CexiCR885p	0.9605334	0.9999529	0.97952664
CexiCR894p	0.97763586	0.999931	0.9666152
CexiCR914p	0.97954845	0.99978966	0.9798422
CexiCR924p	0.9802639	0.9802921	0.9798441
CexiCR927p	0.986859	0.9999529	0.9768921
CexiCR928p	0.97606957	0.9999527	0.9574944
CexiCR938p	0.9765651	0.9999521	0.95060116
CexiCR939p	0.98991215	0.9890963	0.95579463
CexiCR943p	0.9727811	0.9999474	0.9731235

CexiCR957p	0.9805982	0.9968887	0.96731615
CexiCR973p	0.99384594	0.99984705	0.979085
CexiCR1004p	0.9907998	0.99839866	0.972827
CexiCR1016p	0.96387464	0.999953	0.97931194
CexiCR1018p	0.9883702	0.9998897	0.9704902
CexiCR1036p	0.9629518	0.9999529	0.9699409
CexiCR1048p	0.9771141	0.99995023	0.9609422
CexiCR1055p	0.964329	0.99993634	0.957569
CexiCR1060p	0.99363136	0.9998826	0.9825011
CexiCR1066p	0.9798188	0.9999528	0.9810886
CexiCR1070p	0.9862093	0.9999528	0.9664831
CexiCR1074p	0.97263217	0.99995214	0.98129153
CexiCR1096p	0.9905258	0.9999527	0.9587159
CexiCR1117p	0.98037183	0.9961786	0.95103467
CexiCR1146p	0.99396443	0.999953	0.98150426
CexiCR1157p	0.9933666	0.99988014	0.97375035
CexiCR1180p	0.9867725	0.98694044	0.9794615
CexiCR1183p	0.98136294	0.9999503	0.97978234
CexiCR1190p	0.9761144	0.9999523	0.9595231
CexiCR1204p	0.96968937	0.9999409	0.9651625
CexiCR1226p	0.9906999	0.9999525	0.96976113
CexiCR1233p	0.9512744	0.99995273	0.96117425
IscaGr59NC	0.9759698	0.9988481	0.97838295
LhesCR2	0.9914409	0.9508571	0.95994216
LhesCR8	0.9800148	0.9988843	0.97838295
LhesCR46p	0.9853855	0.98268986	0.9612075
LhesCR51p	0.9734181	0.99995244	0.9639706
LpolCR41p	0.9536334	0.9999527	0.97091746
LpolCR50p	0.98804677	0.9989575	0.968979
LpolCR53p	0.9718865	0.99949217	0.9588618

LpolCR60p	0.9623709	0.9999526	0.9788309
LpolCR63p	0.9906679	0.999953	0.98033637
LrecCR103	0.9942692	0.9951453	0.97228575
LrecCR185p	0.9769058	0.99995065	0.98215926
LrecCR189p	0.9890778	0.999953	0.96631134
LrecCR191p	0.9916307	0.9999529	0.9818242
MmarCR3	0.98510027	0.99949247	0.9536712
MmarCR12	0.9911229	0.968042	0.98056126
MmarCR28	0.9917913	0.9766663	0.9575709
MmarCR34	0.9604492	0.9957273	0.97838295
MmarCR40	0.9905736	0.9988166	0.98014367
MmarCR46	0.98022723	0.98090297	0.95415866
MmarCR65	0.9883783	0.9994735	0.97838295
MmarCR86	0.9805781	0.9988587	0.9695988
MmarCR102	0.9814602	0.99984884	0.98242843
MmarCR139	0.9942052	0.99436736	0.9800601
MmarCR148	0.9872519	0.99992275	0.9807596
MmarCR154	0.99303246	0.9639664	0.97235
MmarCR155	0.96643436	0.99908984	0.97915995
MmarCR161p	0.9889339	0.9998993	0.9507644
MmarCR163p	0.98825276	0.9999518	0.96302366
MmarCR164p	0.9853475	0.9999529	0.98112005
MmarCR173p	0.9612085	0.9999527	0.9703734
MmarCR175p	0.972577	0.99994993	0.97978914
MmarCR182p	0.982427	0.9963713	0.9565233
MmarCR220p	0.99504906	0.999951	0.9815037
MmarCR226p	0.9932183	0.9999497	0.97838295
MmarCR235p	0.98499584	0.99995303	0.9808109
MmarCR238p	0.9556836	0.9939568	0.98273695
MmarCR252p	0.96412086	0.9999528	0.9581964

MmarCR258p	0.97632587	0.99995005	0.9756014
MmarCR270p	0.9741979	0.9999311	0.9652635
MmarCR282p	0.9817976	0.99978733	0.9657799
MmarCR284p	0.97853684	0.9999528	0.9799129
MmarCR296p	0.9749657	0.99995273	0.95269036
MmarCR321p	0.9938868	0.9999088	0.9793798
MmarCR323p	0.97757953	0.9999529	0.98140943
MmarCR334p	0.9803846	0.9999492	0.9813953
MmarCR345p	0.9922403	0.99980724	0.9806716
MmarCR356p	0.9653487	0.99995273	0.9803148
MmarCR362p	0.97882605	0.9851196	0.9598742
MmarCR392p	0.99165356	0.999953	0.95966816
MmarCR411p	0.97831595	0.99995285	0.9815134
MmarCR423p	0.99022746	0.99991286	0.95493484
MmarCR425p	0.9898883	0.9999267	0.9790066
MmarCR435p	0.9859898	0.9999529	0.97838295
MmarCR458p	0.98924375	0.999953	0.97866726
MmarCR463p	0.9721862	0.9999517	0.98058236
MmarCR468p	0.96186924	0.99995184	0.96358883
MmarCR486p	0.9641129	0.9999528	0.9697318
MmarCR494p	0.99034375	0.9998703	0.9794482
MmarCR495p	0.98663473	0.9978881	0.96086174
MmarCR507p	0.98910177	0.997242	0.9654912
MmarCR509p	0.9916594	0.984787	0.9839941
MmarCR521p	0.99050117	0.9984281	0.9790528
MmarCR541p	0.98924166	0.9999529	0.9807009
MmarCR548p	0.99314386	0.9999105	0.98025507
MmarCR550p	0.9871918	0.95953643	0.9531789
MmarCR558p	0.9865304	0.99995255	0.9634222
MmarCR575p	0.99162245	0.9999522	0.9552597

MmarCR608p	0.9526217	0.9965713	0.9723517
MmarCR618p	0.9889945	0.9992236	0.96376896
MmarCR644p	0.956803	0.9999528	0.95886564
MmarCR647p	0.97123766	0.9999529	0.98141825
MmarCR650p	0.993469	0.999953	0.98059905
MoccGr12	0.98735815	0.9725559	0.96246445
PtepCR78	0.9942572	0.99884784	0.9741409
PtepCR80	0.9888319	0.98317546	0.9779875
PtepCR103	0.98074114	0.9916123	0.9750015
PtepCR147	0.9738953	0.98675334	0.95913833
PtepCR200	0.98734766	0.99965596	0.9739511
PtepCR378	0.97478175	0.96293193	0.980759
PtepCR381p	0.9828924	0.99995273	0.9619765
PtepCR383p	0.99393797	0.9991542	0.9794456
PtepCR391p	0.99088943	0.9999523	0.98056984
PtepCR408p	0.95749927	0.9999515	0.95429146
PtepCR411p	0.9717957	0.9999529	0.97838295
PtepCR419p	0.9913093	0.99995303	0.97572017
PtepCR420p	0.96567535	0.999953	0.9693273
PtepCR447p	0.97799563	0.9999528	0.9792546
PtepCR448p	0.9603509	0.9999528	0.96716654
PtepCR468p	0.9808657	0.9999529	0.9718616
PtepCR473p	0.99151075	0.999712	0.9626262
PtepCR486p	0.97205496	0.9999523	0.95085484
PtepCR492p	0.98883414	0.9999529	0.96325725
PtepCR499p	0.98575306	0.999953	0.9788078
PtepCR503p	0.9811624	0.9999529	0.9808806
PtepCR516p	0.9923136	0.99995303	0.98166966
PtepCR518p	0.98020923	0.99765146	0.9645369
PtepCR519p	0.9934876	0.9998568	0.979234

PtepCR523p	0.9609287	0.9999523	0.9529482
PtepCR529p	0.98574746	0.99995244	0.9785193
PtepCR530p	0.98082745	0.9999529	0.9572208
PtepCR551p	0.99004686	0.9999529	0.9648925
PtepCR567p	0.9534919	0.999953	0.95256746
PtepCR577p	0.9825603	0.9999517	0.9500091
PtepCR582p	0.98295116	0.999953	0.9789187
PtepCR591p	0.9751463	0.99995303	0.98022735
PtepCR592p	0.971292	0.9999529	0.9800043
PtepCR593p	0.9856056	0.999953	0.9806142
PtepCR594p	0.9911095	0.9999529	0.9803213
PtepCR597p	0.96972805	0.9999529	0.97549534
PtepCR604p	0.9811179	0.9999528	0.96408623
PtepCR617p	0.9804081	0.9999528	0.96026117
PtepCR626p	0.9554074	0.999953	0.9628209
PtepCR628p	0.99196947	0.9999529	0.97838295
PtepCR651p	0.98186123	0.9998594	0.9593385
PtepCR655p	0.97860944	0.9999529	0.982134
PtepCR665p	0.9820081	0.99995303	0.9729558
PtepCR671p	0.96001935	0.98126376	0.9711916
PtepCR678p	0.9926741	0.99991465	0.98028934
PtepCR683p	0.96092397	0.9999529	0.97739446
PtepCR701p	0.9882431	0.99993026	0.97270596
PtepCR706p	0.98714334	0.99995303	0.9804199
PtepCR709p	0.98789847	0.9999522	0.9784288
PtepCR729p	0.98880935	0.9999527	0.9784589
PtepCR745p	0.9880284	0.999953	0.97306466
PtepCR764p	0.9765909	0.9999529	0.96633625
PtepCR769p	0.98968446	0.9999529	0.97496784
PtepCR780p	0.9837061	0.9999527	0.9793419

PtepCR791p	0.9722158	0.99995255	0.97936195
PtepCR805p	0.98294014	0.9999529	0.9759017
PtepCR807p	0.98295903	0.9999529	0.9793043
PtepCR809p	0.9621987	0.99995124	0.9774505
PtepCR817p	0.9896902	0.999953	0.9750277
SmarGr36FI	0.9740904	0.99946713	0.97839034
SmimCR8	0.9879759	0.9542028	0.9683347
SmimCR59	0.98935115	0.9994825	0.9603337
SmimCR147p	0.9892262	0.999953	0.98100436
SmimCR149p	0.9792931	0.9999528	0.9797058
SmimCR153p	0.9851713	0.99995255	0.97361636
SmimCR177p	0.9629438	0.9999529	0.9574554
SmimCR178p	0.98989356	0.9874224	0.98132324
SmimCR183p	0.9861026	0.9997561	0.98037523
SmimCR198p	0.9869218	0.99815667	0.9682429
SmimCR200p	0.987512	0.999866	0.97902876
SmimCR202p	0.98702645	0.9988681	0.9636775
SmimCR212p	0.97995716	0.999953	0.97838295
SmimCR228p	0.9909905	0.999953	0.9804518
TurtCR69	0.98621625	0.9753036	0.9804187
TurtCR152	0.9815021	0.96935	0.9792371
TurtCR208p	0.99150723	0.99995214	0.96856487
TurtCR209p	0.9690256	0.9980639	0.9791796
TurtCR210p	0.95476663	0.99995255	0.958513
TurtCR230p	0.9805553	0.9999529	0.9798147
TurtCR238p	0.977532	0.9999528	0.9797896
TurtCR247p	0.9687119	0.9999527	0.977512
TurtCR252p	0.98659223	0.9999529	0.9797016
TurtCR260p	0.99268126	0.9990295	0.9595978
TurtCR262p	0.97628087	0.9999529	0.96040285