

FACS pipeline - Fast AMP Clustering System

Celio Dias Santos Jr.

2019-07-03

Brief

Fast AMP Clustering System pipeline is a system created by Celio Dias Santos Jr. and Luis Pedro Coelho, from Fudan University (Shanghai / CN). It is distributed under MIT license and represents a new way to prospect AMPs in natural environments using metagenomic data or genomic data to generate large datasets of antimicrobial peptides.

All the references cited along the text are linked to their original publications. To see which papers the text refers to, please click on the year of the publication or the link indicated in the text.

Background

Since the discovery of penicillin and its use in the 1940's, the antibiotics resistance developed by the microorganisms during the medical treatment is a recurrent problem in modern medicine. The fact is to each new discovered antibiotic an emerging resistance trait raises right after six months after their release in market. Part of this is due to the huge metabolism diversity presented by prokaryotes as can be observed in Figure 1. There are two main mechanisms of resistance, one obtained via vertical transference; and the other is action of genes in mobile elements, transmitted both vertically and horizontally to other bacteria. These mobile genetic elements such as plasmids, can carry one or more resistance genes. The prevalent and extremely quick mobility of resistance genes in previously sensitive bacterial populations, now established an world crisis.

The superbugs risen is faster than the time it takes to develop new antibiotics (Figure 2). However, many of these new antibiotics are just chemical modifications of the molecular structure of the old compounds, which makes them prone to be skipped by bacteria using slightly modified strategies. But the question is “can anything be done to slow down the emergence of resistance?”. Antibiotics represent an evolutionary pressure that eventually is the reason to them become obsolete. So, reducing the exposure of microbes to antibiotics can reduce the opportunity for selection and dissemination of resistance. Despite initiatives such as those taken by European Union and in North America, foccus mainly in surveillance and restriction of use. However, these measures are only able to delay the emergence of antibiotcs resistance. Thus, those strategies are welcome, but new drugs will always be needed, since resistance risen is inevitable.

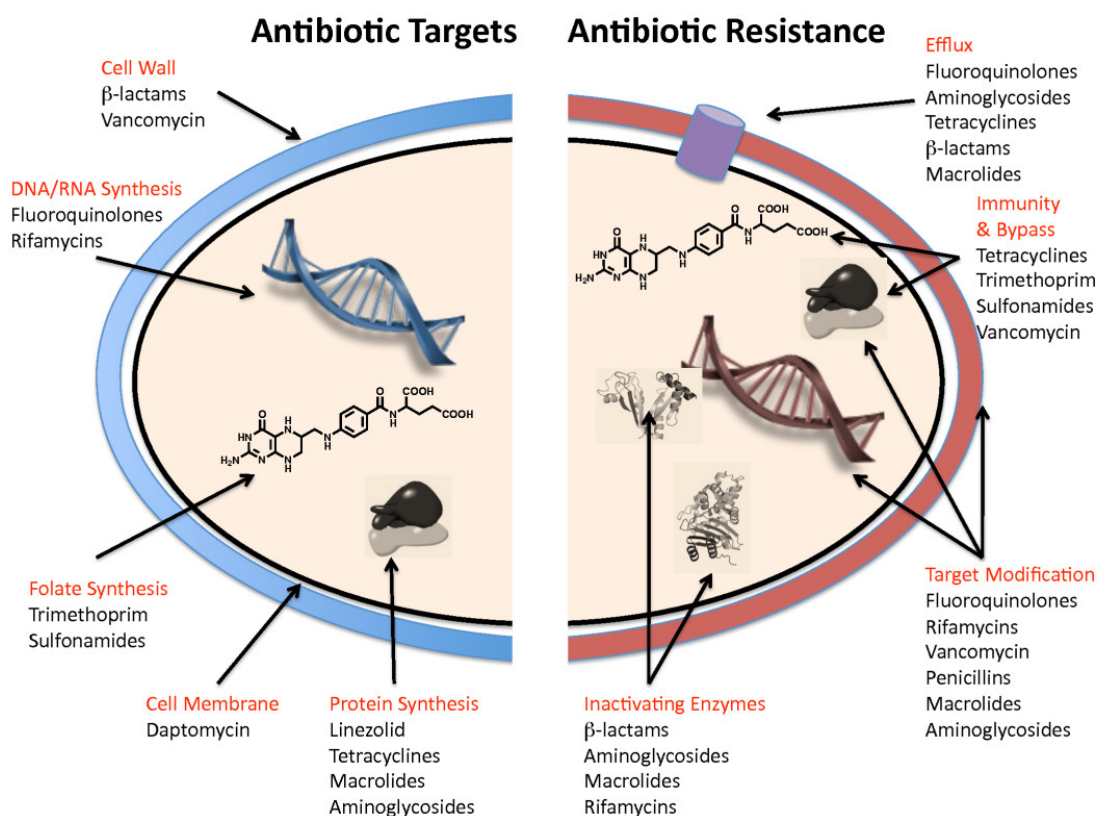


Figure 1. The antibiotics mechanisms and their overcoming (Source: Wright, 2010).

The developing world (Figure 3) usually does not regulate the access to antibiotics and their use in widespread since from agriculture to daily life. This makes antibiotic stewardship an important death cause in those countries, besides the statistics are usually underestimated, since the report in those cases is neglected by many health attendants or the diagnosis is not completed before the patient's death. Rapid intercontinental travels also are efficient to bring pathogens that are no longer geographically contained and can transpose countries easily, like cases involving the spread of the severe acute respiratory syndrome (SARS) virus from Guangdong province in China to Hong Kong and then Canada in 2003.

In a prevision using the growing trend of some diseases that are known to be highly mortal, the resistance to antimicrobials overcome them in 2050, becoming more mortal than cancer (Figure 4). This shows the global importance of the matter, besides to evidence the main countries (Figure 3) affected by this. China is among of the most affected countries and the number of deaths can be higher than 4 million people per year.

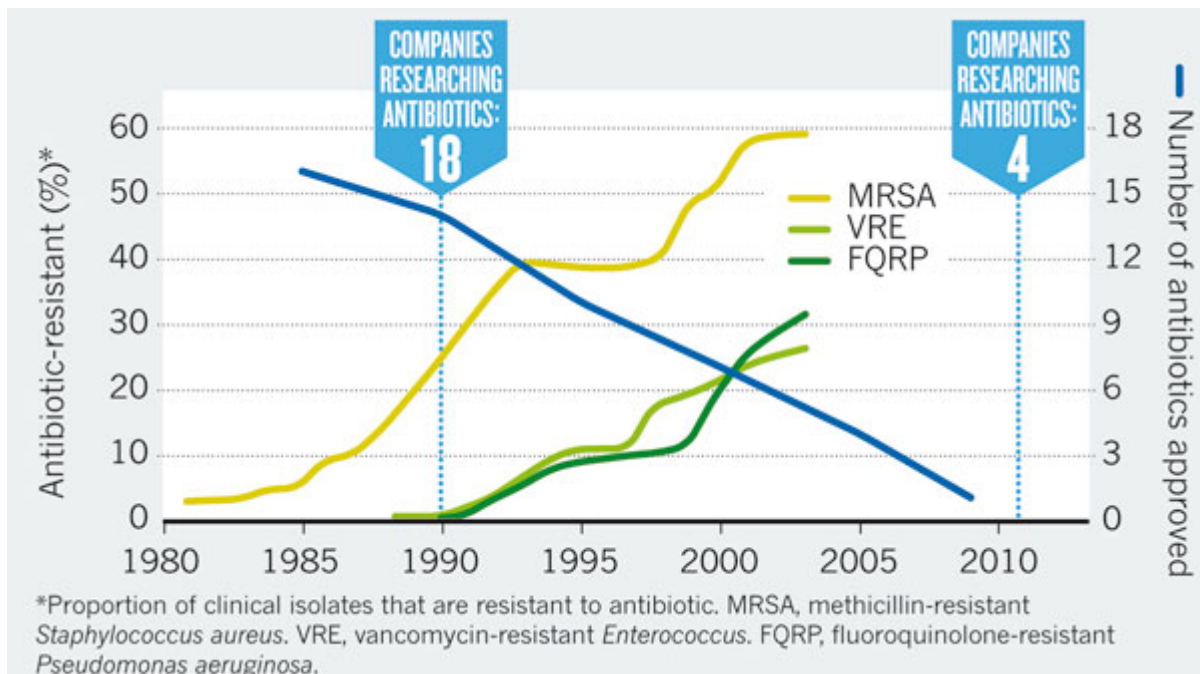


Figure 2. Superbugs running against the pharmaceutical companies in the antibiotics development and overcoming (Source).

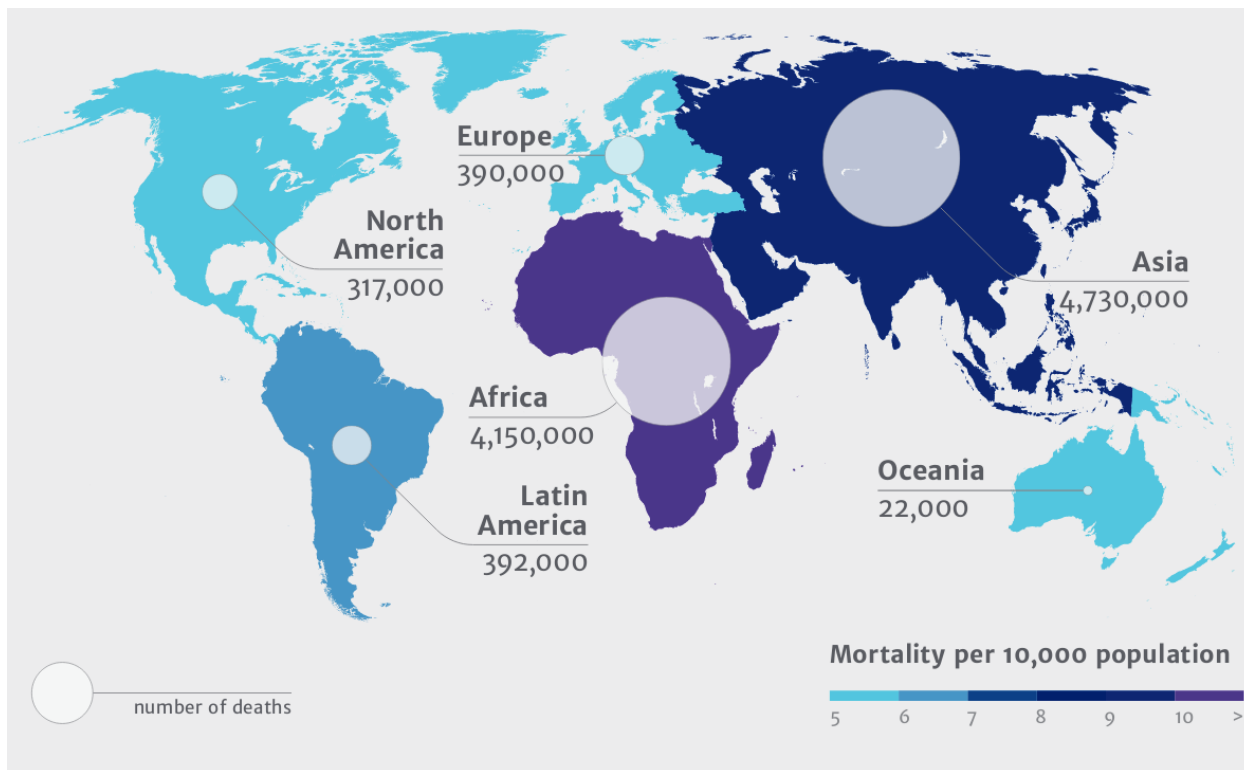


Figure 3. Deaths attributable to antimicrobial resistance every year by 2050 (Source: O'Neil, 2014)

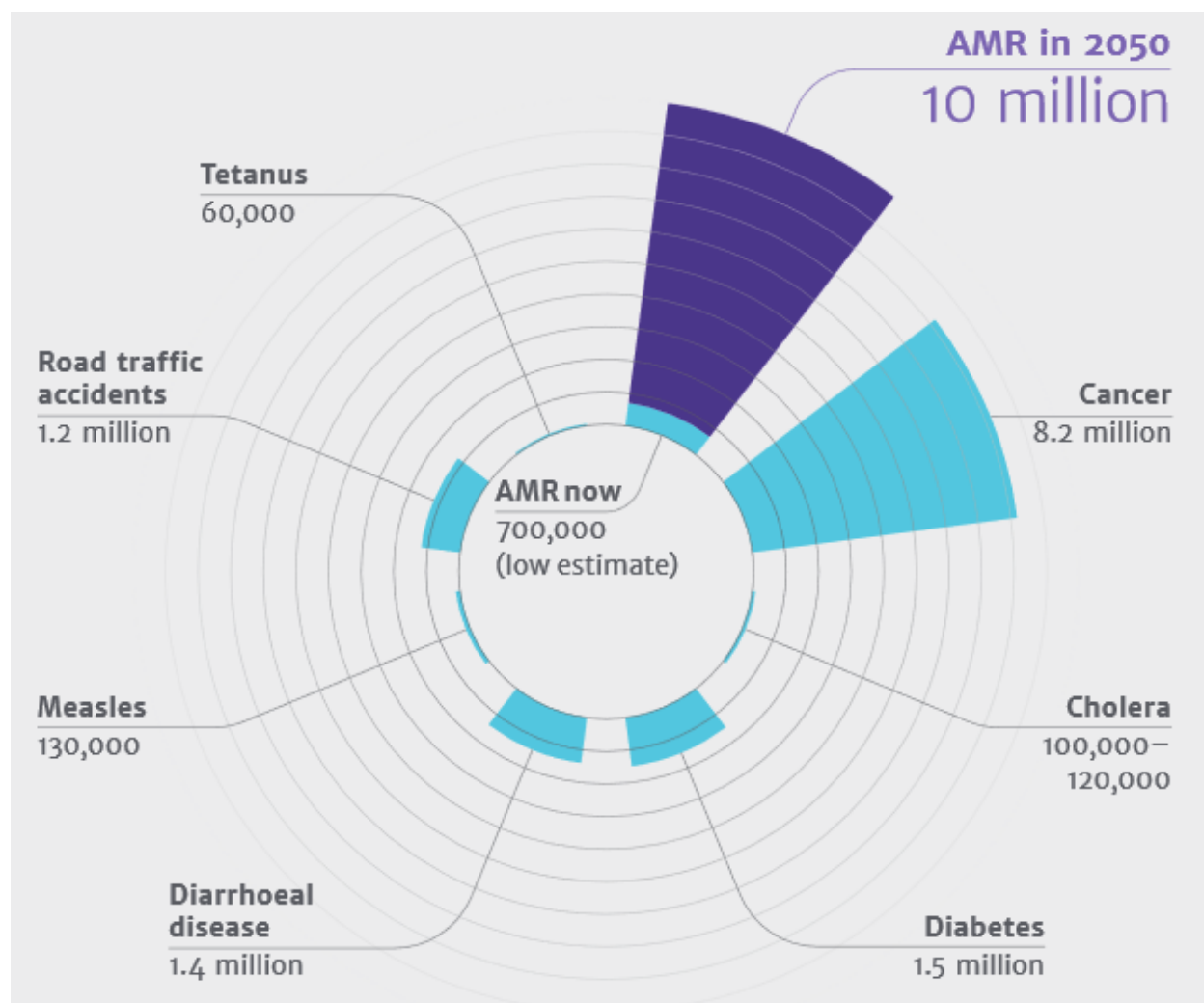


Figure 4. Deaths attributable to antimicrobial resistance every year compared to other major causes of death. (Source: O’Neil, 2014)

Other important side of this problem is the economy. Antibiotic resistance can substantially reduce gross domestic product - but unlike a financial crisis, the damage will last longer and will be greater in the poorest countries (Figure 5). The question here is about the costly treatments and the hospital expenses during treatments of resistance, that can be exhausting and take several months until solved or the patients death.

The growing antimicrobial resistance problem feed a continuous need for new antibiotic drugs, and there are a number of reasons for the scarcity of new antibiotics. Some of them include government regulatory approval adding risk for the pharmaceutical industry, what is usually too criterious, since they will be taken by patients over short periods of time only to cure the disease. Other reason for antibiotic discovery and development decline is scientific. Few compounds discovered with antibiotic properties have had the requisite properties to become drugs. Researchers have argumented that most antibiotics are natural products isolated from soil bacteria, which could suggest the exhaustion of this source now. Many of these ‘natural’ antibiotics have desirable drug-like qualities: good bioavailability, they can cross the cell membrane and have the ability

to evade efflux systems, and chemical structures that favor binding to vital cellular targets. However, there is an increasing difficulty of identifying new chemical compounds with equally suitable drug-like characteristics from natural sources which has caused natural-product-based screening programs disappear in the past few decades. However, the advantages of synthetic compounds are clear to industry, after decades of emphasis on such molecules and millions of dollars spent, no new synthetic antibiotics have emerged.

The genomic era was contrasted by the reality of hundreds of available bacterial genomes that have so far failed to deliver the hoped-for new molecular targets for antibiotics. However, so far it always have focused in the active molecules produced by the metabolism, instead searching for active peptides or proteins. The best reason to bet in host defense antimicrobial peptides or AMPs is that they remained potent for millions of years, constituting a useful strategy to develop a new generation of antimicrobials to meet the growing antibiotic resistance problem worldwide. The current informations about AMPs is extended in that regarding eukaryotes' peptides (Figure 6), and their presence in several phyla in that domain. Although well known in eukaryotes, prokaryotes remain under represented and the few information available does not reflect the entire diversity present in that domain. Archaea is another few explored domain, that can contribute to future drugs development.

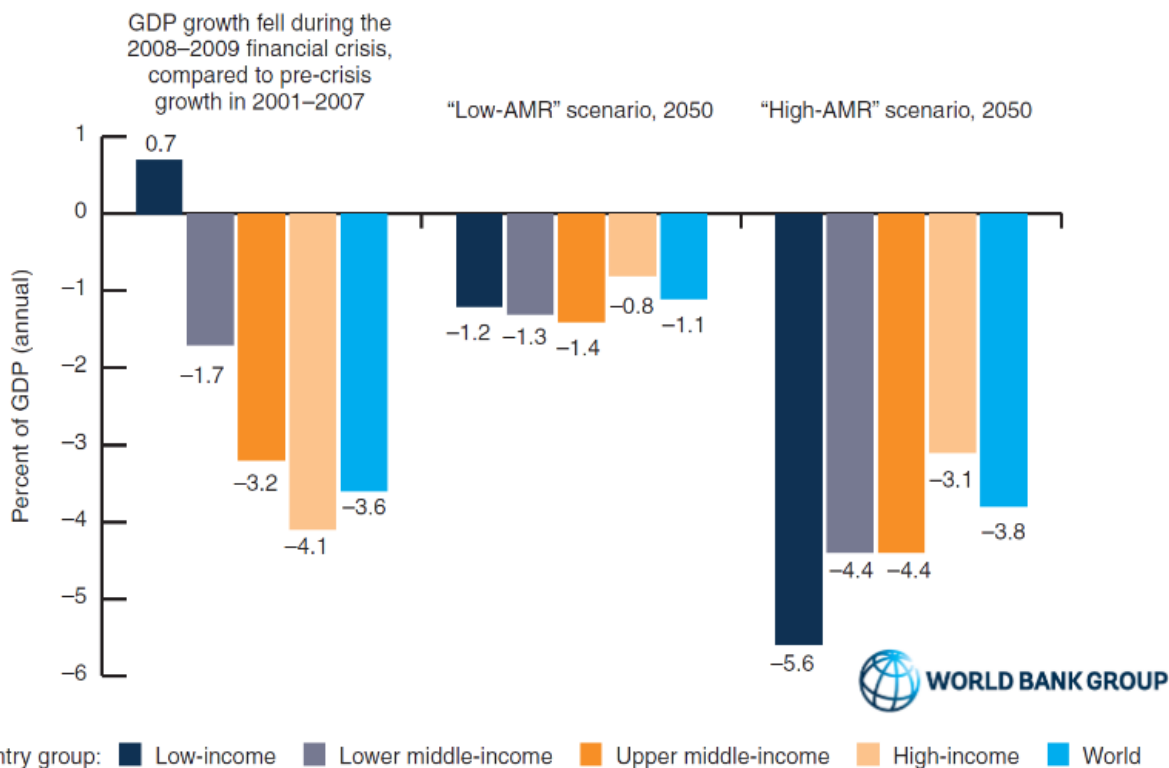


Figure 5. The economy of treatments of antimicrobial resistance cases. (Source: World Bank Group)

AMPs definition comprises peptides with a huge variety of biological activities (Figure 7), and their sequences are key to that activity. AMP producing microbes can limit the growth of other microorganisms and should be considered another normal source of them. AMPs from microbes are quite distinct from those of vertebrates, since they can be obtained from a nonribosomal peptide synthase. Thus, nonribosomal peptides can adopt different structures, such as cyclic or branched structures, and carry modifications like

N-methyl and N-formyl groups, glycosylations, acylations, halogenation, or hydroxylation. Some examples of commercial microbial AMPs include polymyxin B and vancomycin, both FDA-approved antibiotics (Zhang and Gallo, 2016).

Most AMPs are peptides 10-50 amino acids long, they also can reach until 100 amino acids in some cases, with charge ranging between 2 and 11 (some of them being anionic) and constituted of approximately 50% of hydrophobic residues (Zhang and Gallo, 2016). There is a pronounced pH-dependent AMPs charge, mostly resulting in membrane lysis and antibacterial activity at acidic conditions, with many of them not presenting activity at pH higher than 6.0. Thus, the charge seems a key feature in the interaction of AMPs and membranes, where its distribution and nature along the sequence changes the antimicrobial activity (Malmsten, 2014; Pasupuleti et al., 2012; Ringstad et al., 2006). Furthermore, the formation of amphiphilic ordered structures is correlated to peptide-induced membrane disruption. These structures induction, mostly alpha-helices, works as a driving force for membrane binding. Also, the helix destabilization oftenly can reduce the cytotoxicity of AMPs, although this can result in reduction of antimicrobial effects (Malmsten, 2014; Borgden, 2005; Pasupuleti et al., 2012; Hancock and Sahl, 2006; Shai, 2002; Stromstedt et al., 2006).

Antimicrobial peptides from the three domains and five kingdoms of life^a

<i>Domain</i>	<i>Peptide count</i>	<i>Class</i>	<i>Peptide count</i>
Bacteria	209	Insects	216
Archaea	2	Spiders	33
Eukaryota	2,082	Molluscs	27
		Worms	14
<i>Kingdom</i>	<i>Peptide count</i>	Crustaceans	32
Bacteria	209	Birds	36
Protists	7	Reptiles	10
Fungi	12	Fish	79
Plants	301	Amphibians	929
Animals	1,761	Ruminants	44
		Humans	102

Figure 6. Number of antimicrobial peptides found in different domains of life. (Source: Wang, 2014)

Biological activities of host defense antimicrobial peptides

Year created	Activity ^a	Count
2003	Antibacterial (G+/G-)	1,909
2003	Antifungal	850
2003	Antiviral	138
2003	Anticancer	158
2003	Hemolytic	284
2008	Anti-HIV	92
2009	Anti-G+	360
2009	Anti-G-	172
2009	Antiparasitic	59
2009	Insecticidal	22
2009	Spermicidal	9
2011	Chemotactic	47
2012	Anti-protist	4
2013	Antioxidant	10
2013	Anti-inflammatory	2
2013	Wound healing	7
2013	Enzyme inhibitor	5

Figure 7. Antimicrobial peptide biological activities. (Source: Wang, 2014)

AMPs can be classified into 5 families accordingly to their origin and composition (Perumal et al., 2013):

1. Anionic peptides: rich in aspartic and glutamic acids.

Example: Maximimin H5 (from amphibians);

2. Linear alpha-helical cationic peptides: Lack in cysteine.

Example: Cecropins (from insects), dermaceptin (from amphibians);

3. Cationic peptides: rich in proline, arginine, phenylalanine, glycine and thryptophan.

Example: Indolicidin (from cattle), prophenin (from frogs);

4. Anionic and cationic peptides that contain dissulphide bonds: contain cysteine.

Examples: 1 disulphide bond (brevinins),
 2 disulphide bonds (protregrin), and
 3 disulphide bonds (drosomycins and defensins);

5. Anionic and cationic peptide fragments of larger proteins: unusual amounts of W, K, V, R, P, H, L.

Examples: Haemoglobin (from humans), lysozyme,
 ovoalbumin and lactoferricin from lactoferrin.

Families 1-3 are largely found in all domains of life, while families 4 and 5 are more related to eukaryotes and their contribution from microbes is very few (Perumal et al., 2013).

As previously mentioned, these families can be folded into some structural arrangements (Wang, 2014). The most common are shown in Figure 8. The alpha-helical peptides (Figure 8-a) are usually related to a strong pore-forming activity in bacterial membranes, as well as, the alpha-beta structures (Figure 8-c). The beta-sheet peptides usually change their conformation in apolar environments to an alpha-helical structure that can be refolded into beta-sheet (Figure 8-b) after transposition of the lipophilic phase. The random coiled peptides (Figure 8-d) are usually associated to a mixed function, and usually assume helical structures in the membrane, forming pores and compromising cell functions. Mostly the activities of AMPs are associated to the rupture of cell membrane or promoting the leakage of cell contents, ending in the bacterial cell death. Different biological activities have different mechanisms, however in this review the antibacterial activity will be prioritized.

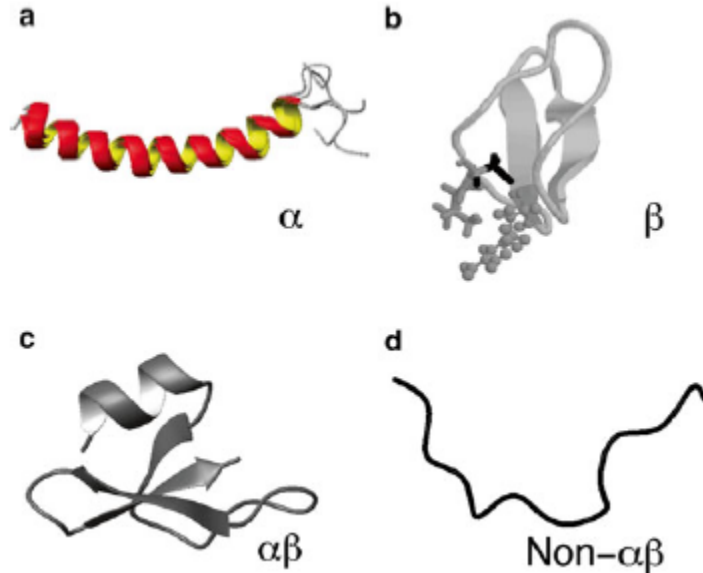


Figure 8. Antimicrobial peptide folding groups. (Source: Wang, 2014)

There is a dynamic interchange in AMPs structure and topologies along the interaction with the microbial cell membranes (Samson, 1998). The outer surface of prokaryotic cells is negatively charged (mainly due to lipopolysaccharides and teichoic acid), what promotes an electrostatic interaction of AMPs with the membrane being the primary mechanism for antimicrobial activity. Other cases are based in the AMP

translocation across the cell membrane and the inhibition of essential cellular processes (e.g. protein synthesis, nucleic acid synthesis, enzymatic activities) (Brogden, 2005). Based on the mechanisms of action, AMPs are categorized into membrane acting and nonmembrane acting peptides. The first ones are capable of forming transient pores on the membrane, whereas the second ones have the ability to translocate across the cell membrane without permeabilizing it (Pushpanathan et al., 2013).

Several models have been proposed to describe the mechanism of action of antimicrobial peptides (Figure 9), and can be categorized into energy dependent and energy independent uptake. In barrel-stave mechanism, there is an aggregation of peptide monomers on the surface of the membrane. This aggregated peptides are inserted into the membrane and get such an orientation that the hydrophilic surfaces of peptides point inward and form a water filled transmembrane pore that kills the cell by leakage. In carpet model, AMPs initially get associated on the surface of the membrane, forming a carpet. Once a concentration reaches a threshold, there is a peptide induced membrane permeation. This leads to the cell membrane disruption. In toroidal pore model, peptides get aggregated prior or after binding with the membrane surface. It induces a membrane depolarization and form a toroidal shaped transmembrane pore. The energy independent uptake involves macropinocytosis. Once uptaken in the form of macropinosomes, the AMPs get released into the cytoplasm exerting their antimicrobial action (Pushpanathan et al., 2013).

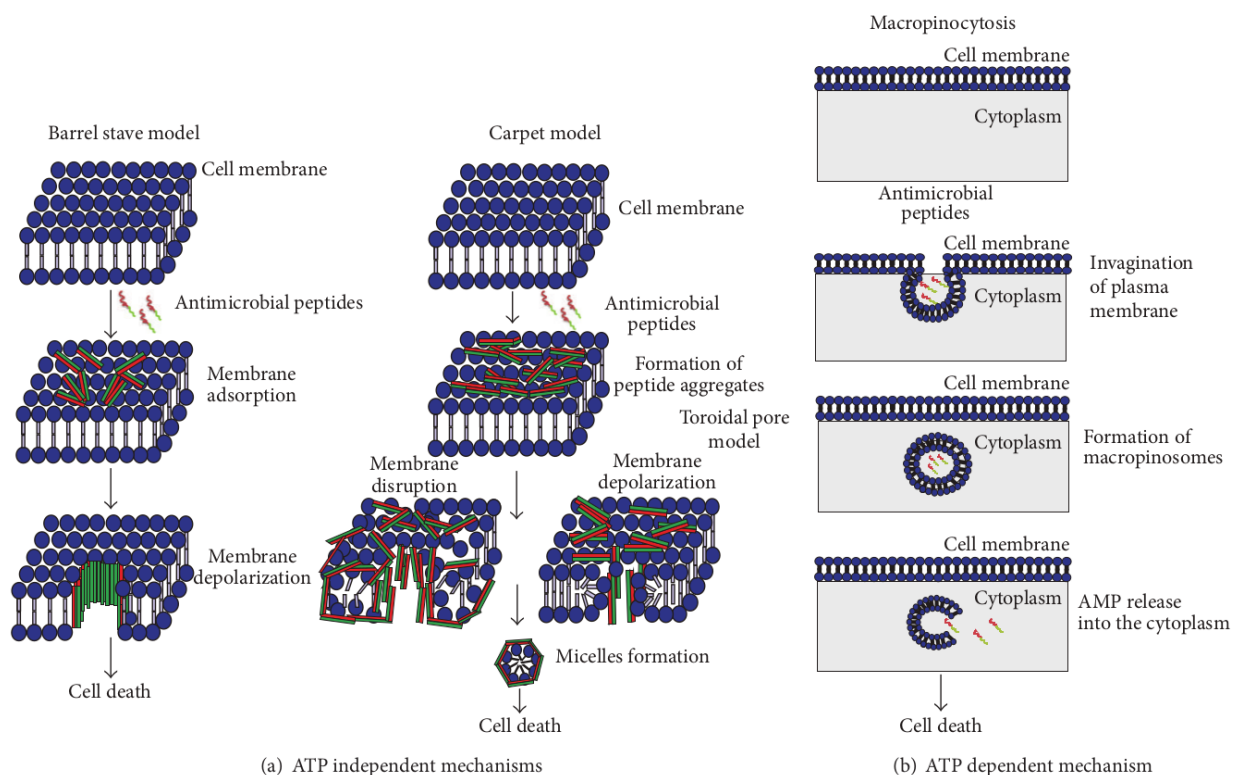


Figure 9. Proposed mechanisms of actions of AMPs: Energy independent mechanisms - barrel stave model, carpet model, and toroidal pore model (a); and energy dependent mechanisms (b). (Source: Pushpanathan et al., 2013)

Microbes were thought to be unable to develop resistance towards AMPs. However, recently some resistance mechanisms have been reported, such as upregulation of proteolytic enzymes able to degrade AMPs,

membrane modifications resulting in decreased negative potential of bacterial membranes, and release of glucose aminoglycans, polysaccharides, and other polyanionic species able to scavenge AMPs (Nizet, 2006). Despite having been convincingly demonstrated *in vitro*, resistance development to AMPs *in vivo* needs to be further clarified, since conditions experienced by bacteria in a laboratory setting are likely to differ from those *in vivo*. In the latter case, the microbes are exposed to a cocktail of AMPs, which may reduce or alter the selection pressure underlying resistance development (Malmsten, 2014).

In summary, the AMPs represent a multidimensional group of molecules with several applications, among them:

- Drug delivery vectors
- Mitogenic agent
- Antitumour agent
- Signaling molecules
- Contraceptive agent for vaginal prophylaxis
- Plant Transgenesis

Our main goal with FACS is a highthroughput screening system of AMPs classifying them into the main classes accordingly to their electrostatic nature and 3-D structure, also regarding their natural propensity to cause hemolysis or not. These data can be used in several future applications, such as a catalogue to biotechnological production or screening of activity and description of environments or conditions related to the microbial population regulations.

Pipeline overview

FACS is a pipeline to:

1. merge paired-end reads,
2. predict peptides,
3. cluster them at 100% of similarity and 100% coverage,
4. calculate their abundance in peptides per million (ppm), and
5. select those with antimicrobial potential discriminating their hemolytic potential.

With FACS you can treat a metagenome file of 631.2Mbp as fast as 24 min, using 3 cpus and 100Mb sequence buckets in a Ubuntu v.18 64x bits.

FACS pretty much works in three main steps (Figure 10). The first step is about ordering the paired-end reads of the files forward and reverse by name, also realizing a quality trimming and eliminating orphan reads. This step is crucial to ensure the reads will be sorted correctly to be merged into longer quasi-contigs. These sequences then are screened by ORF sequences with minimum of 30 base pairs. This ensures the AMP

sequences are being sampled and also eliminates short peptides without possibility to be selected in the next steps for being too short. Then, these ORFs are ordered by sequence and the redundant sequences with 100% of identity and 100% of coverage are collapsed and this number is counted. The abundance of the peptides are then calculated by summing the total of detected peptides and dividing the occurrences by this total and after multiplying it by $1e6$. This abundance measure is given as “ppm” (peptides per million). During this process of clustering the sequences are divided into sequence buckets of a customizable size that is related to the total amount of RAM memory available in the computer. It can help to speed up the process and allows large datasets being processed without memory errors. These buckets are then used in the downstream operations until the end of the program when the informations are gathered in a final table.

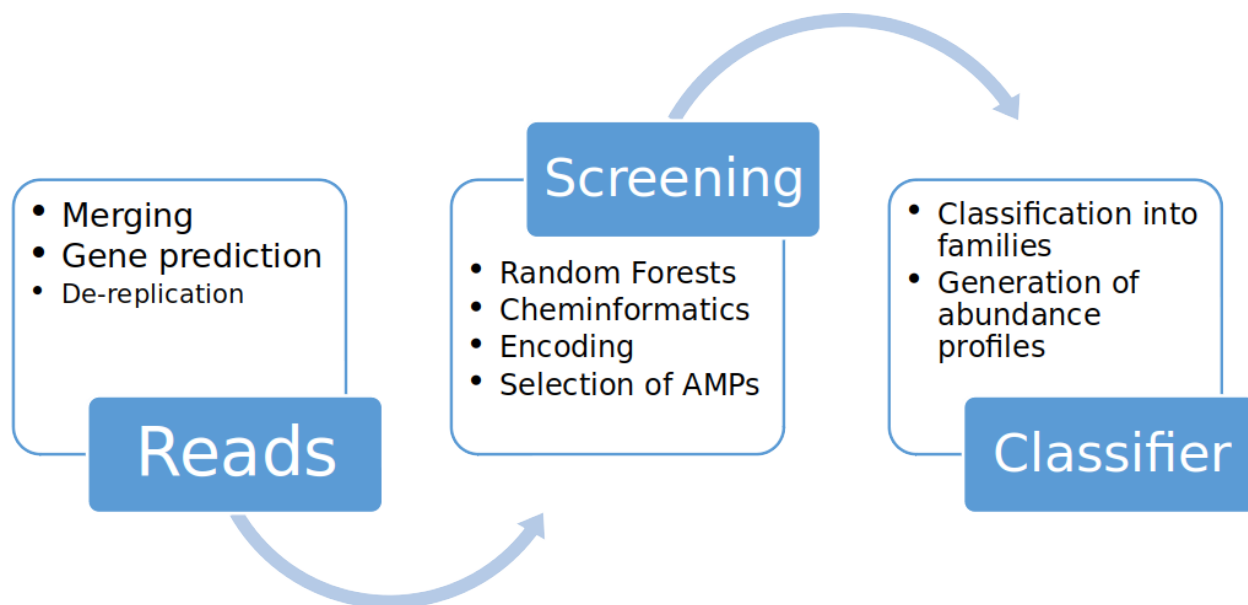


Figure 10. FACS workflow.

The second step is about calculation of descriptors. In FACS a two way system of descriptors was adopted using cheminformatics allied to sequence encoding, since the both methods were previously applied to AMP screening, but we believe there is a synergy of those informations. We will further discuss about the descriptors used in FACS in detail. The descriptors calculation is made entirely in R using subscripts that run along the FACS pipeline using the sequence buckets and returning only AMP sequences that are then classified into hemolytic or non-hemolytic. The classifiers adopted in FACS are based in random forest algorithms (further discussed in details later) that proved to be more efficient than those previously reported (Gabere and Noble, 2017; Bhadra et al., 2018, Meher et al., 2017). Finally, in the third step FACS performs a classification using a decisions tree (Figure 11) that classifies the detected AMPs into four different families accordingly their nature (Cationic or Anionic) and structure (linear or disulphide bond forming). These classifications are then made available in a table where the sequence, random identifiers, abundance in ppm and hemolytic nature is also added. Interestingly, the FACS workflow depends on few third party softwares and some R libraries (Figure 12).

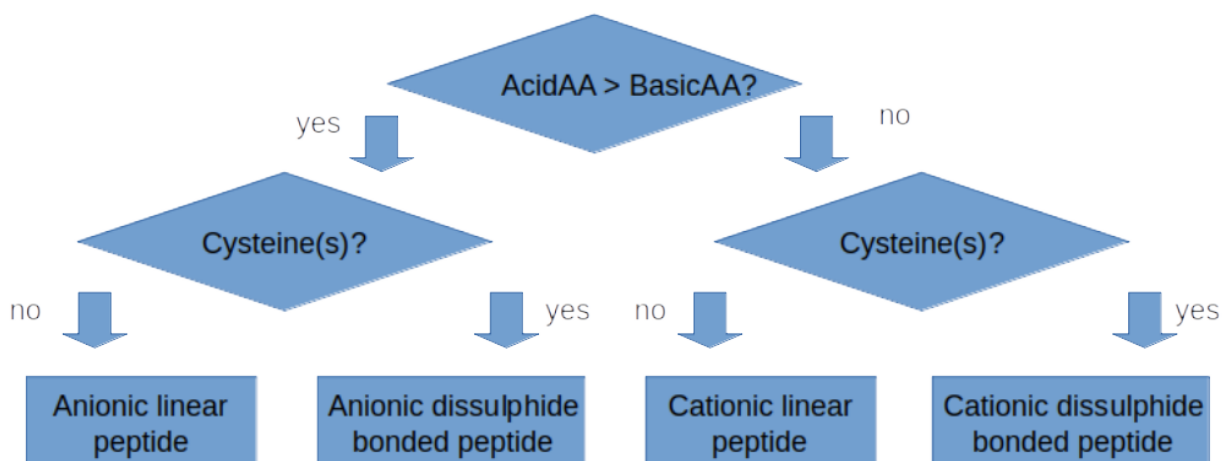


Figure 11. Decision tree to classification of peptides into different classes accordingly to their composition and capacity in forming dissulphide bonds (Legend: AcidicAA - Acidic amino acids: B + D + E + Z; BasicAA - Alkaline amino acids: H + K + R).

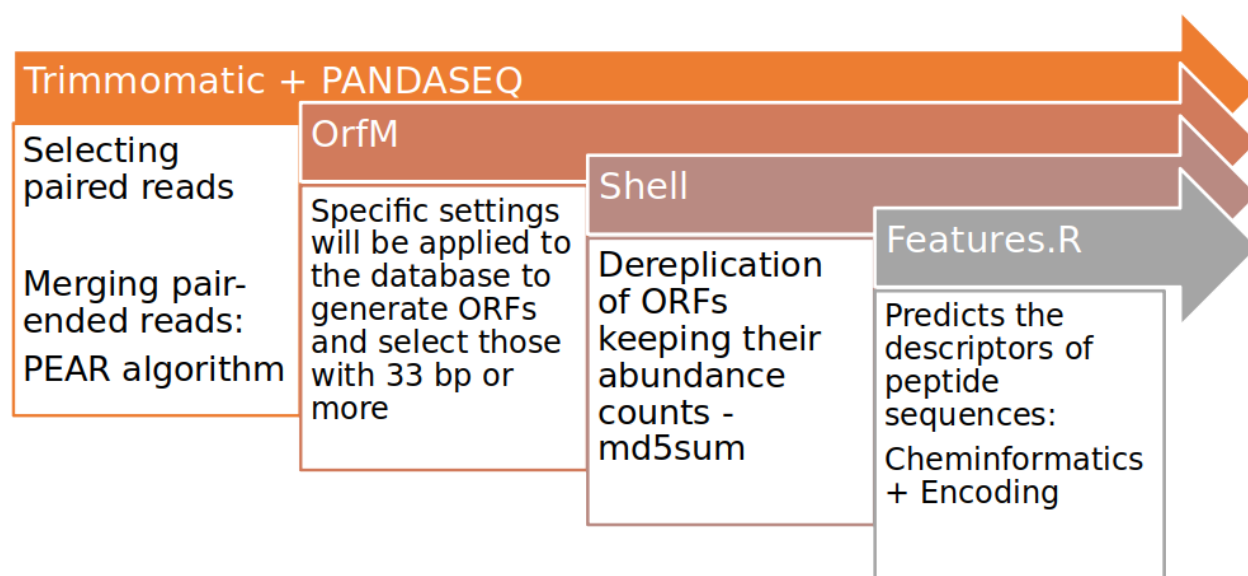


Figure 12. FACS structure.

In general FACS make use of the pigz software to compress and decompress files quickly, as well as GNUParallel to make the shell processes faster and parallelized. In order to make it clear, the processes are explained into small single steps:

1. Reads sorting and trimming (Trimmomatic);
2. Reads merging (Pandaseq);
3. ORFs prediction (ORFm);
4. Sorting of sequences using sort with memory options;

5. Clustering using uniq -c and memory options;
6. Abundance is calculated mainly by awk functions;
7. Descriptors calculations is made by using R scripts that relies on R packages: Peptides, data.table, dplyr, parallel, doParallel;
8. AMPs prediction and Hemolytic activity classification as well as the families identification is basically implemented in R language and uses mostly the following R packages: randomForest, caret, data.table, dplyr;
9. The final formatting of files is performed by shell functions.

Descriptors system: Distribution

Spänig and Heider (2019) recently released a series of sequence encoding methods and a review of the main machine learning models using them. For a better comprehension of the following topics, we strongly recommend the its reading.

A recent prediction method released by Bhadra et al. (2018), has shown that sequence encoding methods are enough to a good classification of AMPs in a large dataset of peptides. However, the proportion of AMPs to non-AMPs in the dataset influenced the results of the classifier. In this sense, they used a total of **23** different descriptors mostly based in the CTD method (Figure 13).

The CTD method was firstly described by Dubchak et al. (1995,1999) and it is based in the classification of residues into three different classes accordingly to some specific features, such as hydrophobicity, solvent accessibility or secondary structure. The peptide sequence then is encoded into these three classes and the composition, distribution and transition of classes can be calculated. Mostly the composition and transition are important to other applications than AMP prediction, since they have shown small correlation to AMP peptides, besides not being explicative in some tested models performed by us previously.

The CTD descriptors are developed by Dubchak *et al.* (1995) and Dubchak *et al.* (1999).

Sequence	M	T	E	I	T	A	S	M	V	K	E	L	R	E	A	T	G	T	G	A
Sequence Index	1				5					10					15					20
Transformation	3	2	1	3	2	2	2	3	3	1	1	3	1	1	2	2	2	2	2	2
Index for 1			1							2	3		4	5						
Index for 2		1			2	3	4								5	6	7	8	9	10
Index for 3	1			2				3	4			5								
1/2 Transitions																				
1/3 Transitions																				
2/3 Transitions																				

Figure 13. Method of sequence encoding using CTD (Composition, Distribution and Transition). (Source: Dubchak et al., 1995)

As above mentioned, since distribution of the residues classes seemed to be more effective to explain and classify AMPs, AMPEP software (Bhadra et al., 2018) was mostly based in the distribution of the classes of the five canonical features (hydrophobicity, normalized van der Waals volume, polarity, polarizability, charge, secondary structure, and solvent accessibility), as shown in Figure 14.

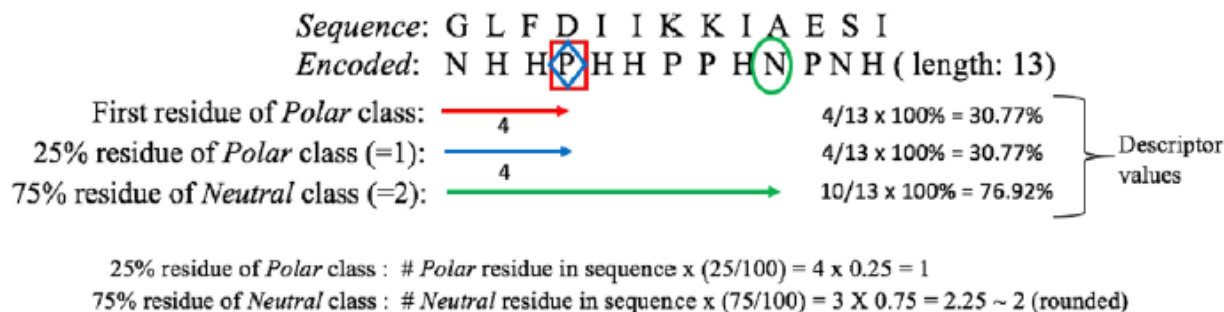


Figure 14. Example of CTD application to AMP discovery in AMPEP software. (Source: Bhadra et al., 2018)

However, Bhadra et al. (2018) observed that the distribution was more important when taken from the first residue, calculated as:

$$Z = [R \times Y / 100], \text{ where:}$$

- R is the total number of class residues in the sequence,
- Y denotes the desired percentage.

In this sense, despite the high Accuracy and sensitivity, other works still suggest that methods independent of sequence order and mostly based in cheminformatics have with comparable statistics (Boone et al., 2018). Thus, Fjell et al. (2009) has shown using a combination of 77 QSAR (quantitative structure-activity relationships) descriptors that artificial neural network models could predict the extension of peptides activity, not only classify them.

Thus, these methods could be joined to achieve a better performance and also economy of computational resources, since sequence encoding represents a considerable cost of processing. The alliance between those two methods can fix their pitfalls, since the sequence order independent methods fail in classify, however are good to describe activity; and sequence encoding is essential to a good classification, but fails when predict activity extent.

The descriptors adopted by FACS are hybrid being partially cheminformatics and sequence encodings, what by itself represents a breakthrough. FACS performs firstly a distribution analysis of three classes of residues in two different features (Solvent accessibility and *Free energy to transfer from water to lipophilic phase*) as shown in Table 1.

Table 1. Classes adopted to the sequence encoding of the distribution at Residue0. The Solvent Accessibility was adopted as other studies previously Dubchak et al. 1995, 1999, however the new feature “Free energy to transfer to lipophilic phase” was adopted from Von Heijne and Blomberg, 1979.

Properties	Class I	Class II	Class III
Solvent accessibility	A, L, F, C, G, I, V, W	R, K, Q, E, N, D	M, S, P, T, H, Y
FT	I, L, V, W, A, M, G, T	F, Y, S, Q, C, N	P, H, K, E, D, R

The novelty in this method is use the *Free energy to transfer from water to lipophilic phase* (FT) firstly described by Von Heijne and Blomberg, 1979. This measure is based in the estimation of free energy difference for the transfer of a residue from a random coil conformation in water to an alpha-helical conformation in a lipophilic phase (membrane). FT is calculate taking in account hydrophobicity, charge and polarity of the residues what reduces 3 features used to calculate CTD to one. Besides that, it also seems much more credible and important to predict AMPs since their functions are mostly based in the membranes interaction, and FT seems a measure of the potential of peptide insertion into them. To build the three classes, FT measures by residue were normalized as Zeta-Scores and then sorted into three groups (Table 1). From the distribution of those three classes, FACS uses the first residue measure, getting the following descriptors:

- SA.G1.residue0
- SA.G2.residue0
- SA.G3.residue0
- hb.Group.1.residue0
- hb.Group.2.residue0
- hb.Group.3.residue0

The other cheminformatic descriptors are widely used in the AMPs description, and follows:

- tinyAA ($A + C + G + S + T$)
- smallAA ($A + B + C + D + G + N + P + S + T + V$)
- aliphaticAA ($A + I + L + V$)
- aromaticAA ($F + H + W + Y$)
- nonpolarAA ($A + C + F + G + I + L + M + P + V + W + Y$)
- polarAA ($D + E + H + K + N + Q + R + S + T + Z$)
- chargedAA ($B + D + E + H + K + R + Z$)

- basicAA (H + K + R)
- acidicAA (B + D + E + Z)
- charge (pH = 7, pKscale = “EMBOSS”)
- pI (pKscale = “EMBOSS”)
- aindex (relative volume occupied by aliphatic side chains - A, V, I, and L)
- instaindex -> stability of a protein based on its amino acid composition
- boman -> overall estimate of the potential of a peptide to bind to membranes or other proteins as receptor
- hydrophobicity (scale = “KyteDoolittle”) -> GRAVY index
- hmoment (angle = 100, window = 11) -> quantitative measure of the amphiphilicity perpendicular to the axis of any periodic peptide structure, such as the alpha-helix or beta-sheet

These descriptors are used to prediction and are calculated to each sequence that was identified as a potential peptide.

Datasets and training

In a recent study, Gabere and Noble (2017) have shown that Random Forests models provide a statistically significant improvement in performance of AMPs detection, as measured by the area under the receiver operating characteristic (ROC) curve in comparison to other methods. Following this trend, many classifiers, such as AM Pep and others, also make use of random forests models and presented highly accuracy in AMPs detection. Due to this, we opted to use random forests after some tests using alternative algorithms, such as: treebag, rpart, cpart, adaboost and others (results not shown). In order to be able to trace a comparison among the other classifiers and our model, we opted to use the same training and validation datasets used by Bhadra et al. (2018), shown in Figure 15.

Dataset	Model Design	Comparative Study	
	Training ($M^{\text{model_train}}$)	Benchmark Training (C^{train})	Benchmark Testing (C^{test})
Positive	APD3, CAMPR3, LAMP {3268}	Xiao {770}	Xiao {920}
Negative	UniProt {166791}	Xiao {2405}	Xiao {920}

Figure 15. Dataset for training and validation of antimicrobial peptides prediction model. (Source: Bhadra et al., 2018)

The 22 descriptors were generated by using the customized script written for this purpose implemented in FACS. These descriptors were organized into tables containing 26 columns, where the first three stood for: peptide access code, sequence and abundance. Then, two models were generated by training the AIs with the small and large datasets. Models were trained using random forest R package with a 10-cross fold validation and auto-mtry. The final models were tested by prediction internal system, being validated against the validation dataset containing AMP:non-AMP proportion of 1:1, and a total of 1840 peptides (Figure 15).

The hemolytic activity classifiers was obtained by using the datasets previously established by Chaudhary et al. (2016). The HemoPI-1 datasets were used both to training and validation (Figure 16). Following the same pre-established logic, the descriptors were calculated as the standard method implemented in FACS by using Peptides R package and CTDDclass.py script from ILearn project, already implemented by the installation procedures bellow further discussed.





HemoPI-1 Datasets			
Dataset	Description	Positive	Negative
MAIN	It contains 442 experimentally validated hemolytic peptides (positive examples) from Hemolytik Database and 442 randomly generated peptides from Swiss Prot as non-hemolytic.		
VALIDATION	It contains 110 experimentally validated hemolytic peptides (positive examples) from Hemolytik Database and 110 randomly generated peptides from Swiss Prot as non-hemolytic.		

Figure 16. Dataset for training and validation of hemolytic peptides prediction model. (Source: Chaudhary et al., 2016)

Models for hemolytic activity were trained using caret R package, using the 22-descriptors dataset with the algorithm for Oblique Random Forests using Support Vector Machines (orfSVM). Training was performed by using a 5-cross fold validation repeated 3 times. This algorithm is a breakthrough in the predictions, since Chaudhary et al. (2016) used SVM and got similar results. However, differently from their work, this model works faster using much less descriptors and is implemented in R language, not Java, which was highly dependant of the users settings and not portable after all.

The models here mentioned were implemented in a R script to filter off the non-AMP peptides and classify AMPs into hemolytic or not. After that, this very script also makes the decisions tree sorting presented in Figure 11, classifying AMPs also into the 4 families:

- Anionic linear peptides (ALP)
- Anionic disulphide-bond forming peptides (ADP)
- Cationic linear peptides (CLP)
- Cationic disulphide-bond forming peptides (CDP)

Models and prediction accuracy

The classifiers of AMP and hemolytic peptides were then assessed and compared to the state of art methods of AMP prediction and hemolytic classification. The results were taken using the same databases of the works cited as reference. Calculations of the statistics of accuracy, sensitivity, McNeymar's Correlation Coefficient (MCC), and F-score were shown in Figure 17.

$$Sn = \frac{TP}{TP + FN} \quad (1)$$

$$Sp = \frac{TN}{FP + TN} \quad (2)$$

$$Pr = \frac{TP}{TP + FP} \quad (3)$$

$$Acc = \frac{TP + TN}{TP + FN + FP + TN} \quad (4)$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP) \times (TP + FN) \times (FP + TN) \times (TN + FN)}} \quad (5)$$

Figure 17. Measures of model accuracy. Legend: FN - False Negatives, TP - True Positives, TN - True Negatives, FP - False Positives. (Source: Bhadra et al., 2018)

The AMP prediction model was compared at two levels the first level with it trained with the small training dataset and when trained with the AMPEP complete dataset, presenting a ratio of 1:3 (positives:negatives). The final results (Table 2) shows clearly that R22 models are comparable efficient in retrieving AMPs from the validation dataset reaching accuracies very close to the best system so far (AMPEP). However, R22 trained with the complete dataset (R22_LargeTrainingset) outperforms AMPEP with a better accuracy, specificity and precision. These features also reflects a better global adjustment of this model, since its MCC and F-score were higher than those from AMPEP. Thus, R22_LargeTrainingset seems the best model to be used in the AMPs prediction, besides use less descriptors and memory. Besides, It also is all implemented in R, what gives portability to the process that could be entirely implemented in FACS just by external scripts, composing a bigger workflow.

Table 2. Models to predict antimicrobial peptides were tested to benchmark the results obtained with the new set of descriptors adopted in this classifier. All models and systems were tested with the benchmark validation dataset from the AMPEP study available in Bhadra et al. (2018).

Model/Method	Acc.	Sp.	Sn.	Precision	F-Score	MCC	Reference	Descriptors
AMPep	0.962	0.965	0.95	0.913	-	0.9	doi:10.1038/s41598-018-19752-w	23
Peptide Scanner v2	0.755	0.686	0.943	0.523	0.673	0.557	doi:10.1093/bioinformatics/bty179	-
CAMPr3-NN	0.728	0.704	0.794	0.494	0.609	0.445	doi:10.1093/nar/gkv1051	-
CAMPr3-RF	0.584	0.461	0.923	0.384	0.543	0.354	doi:10.1093/nar/gkv1051	-

Model/Method	Acc.	Sp.	Sn.	Precision	F-Score	MCC	Reference	Descriptors
CAMPr3-SVM	0.6	0.506	0.858	0.388	0.534	0.328	doi: 10.1093/nar/gkv1051	-
CAMPr3-DCA	0.617	0.542	0.821	0.396	0.534	0.324	doi: 10.1093/nar/gkv1051	-
iAMP	0.548	0.413	0.918	0.363	0.521	0.313	doi: 10.1038/srep42362	-
AMPA	0.779	0.941	0.336	0.675	0.449	0.361	doi: 10.1093/bioinformatics/btr604	-
R22_Ctrained	0.952	0.972	0.932	0.971	0.951	0.904	This study	22
R22_Large	0.967	1	0.934	1	0.966	0.936	This study	22

The specific results of the confusion matrix are presented now (Table 3) to the R22_LargeTrainingset. The 100% of specificity does not mean an overfitting since there is a clear misclassification of positive peptides, which also ensures that the model is still reliable to be used in other datasets.

Table 3. Confusion Matrix of R22_LargeTrainingset model.

Prediction/Reference	AMP	Non-AMP
AMP	859	0
Non-AMP	61	920

Meanwhile, the hemolytic prediction model was evaluated using its own datasets and trained as previously informed. The results of this model and the comparisons to the standard system currently available are shown in the Table 4. The model obtained with oblique random forests supported by vector machines was a bit less accurate, however the sensitivity and specificity were higher than that obtained previously. Moreover, the MCC measure also shows our model performance similar to the best model currently implemented in the server, based in support vector machines (SVM). The similarities among their performances are important to make sure our model is reliable, regarding the convenience of being implemented in R with a set of descriptors previously calculated to the AMP prediction model. In this way, the previous tables can be reused in this case, saving time and memory.

Table 4. Models used to predict hemolytic activity were trained with the same dataset used by Chaudhary et al. 2016 and were tested with the same test dataset used by them to benchmark the results obtained with another model used to generate this classifier.

Methods	Study	Sn (%)	Sp (%)	Acc (%)	MCC
SVM	Chaudhary et al., 2016	95.7	94.8	95.3	0.91
IBK	Chaudhary et al., 2016	95.5	93.7	94.6	0.89
Multilayer Perceptron	Chaudhary et al., 2016	93.9	92.8	93.3	0.87
Logistic	Chaudhary et al., 2016	93.4	93.7	93.6	0.87
J48	Chaudhary et al., 2016	89.6	88.5	89.0	0.78
Random Forest	Chaudhary et al., 2016	94.1	94.6	94.3	0.89
ORFsvm	This study	95.5	95.5	95.5	0.91

Our classifiers seems to be extremely interesting in the execution of the filtering off non-AMP peptides and classifying them into hemolytic or non-hemolytic peptides. Also, the models were implemented in the same programming language (R) and used the same set of descriptors, what saved time and memory in the process implemented in FACS.

As a future update to FACS classifier systems, some efforts recently have being done, in order to achieve a model to classify the biological activity presented by AMPs: anti-bacterial, anti-fungal, anti-viral, anti-HIV and anti-tumor. A test was carried out using the same set of descriptors and training with the AMPEP training dataset available here. The training procedures were same adopted before to both models, testing the random forest (rf) and the oblique random forests with support vector machines (orfsvm) algorithms. The model here presented is a result of the 10-cross fold validated random forest training and showed a limited capacity of classification (Figure 18), with a specificity and sensitivity ranging to very low values in some classes.

Overall Statistics						
Accuracy : 0.9239						
95% CI : (0.9151, 0.9321)						
No Information Rate : 0.6181						
P-Value [Acc > NIR] : < 2.2e-16						
Kappa : 0.8652						
	Class: antibac	Class: antifungal	Class: antihiv	Class: antitumor	Class: antiviral	Class: NAMP
Sensitivity	0.9740	0.77869	0.53488	0.57143	0.233871	1.0000
Specificity	0.9740	0.97078	0.98476	0.99440	0.991240	1.0000
Pos Pred Value	0.9025	0.73454	0.44231	0.79208	0.467742	1.0000
Neg Pred Value	0.9935	0.97688	0.98944	0.98417	0.975189	1.0000
Prevalence	0.1979	0.09406	0.02210	0.03598	0.031868	0.6181
Detection Rate	0.1928	0.07325	0.01182	0.02056	0.007453	0.6181
Detection Prevalence	0.2136	0.09972	0.02673	0.02596	0.015934	0.6181
Balanced Accuracy	0.9740	0.87473	0.75982	0.78292	0.612555	1.0000

Figure 18. Confusion matrix of model “Mixedmodel_multiclassifier”. This model tried to classify the 8 different AMP biological activities. To train it we have used the large training dataset, previously provided by Bhadra et al., 2018.

While this classifier works well for prediction of anti-bacterial activity with (sensitivity and specificity of ~97%), other activities, such as anti-viral activity is badly classified with a sensitivity of only 23%. This reflects, in a general way, a non-homogeneous performance, also regarding the biochemical diversity of the peptides present in each one of those classes. These results suggest that a bigger training set, well annotated and maybe another conditions of training could improve the prediction performance.

Testing

FACS was further tested in two different metagenomes SRR90186022 and SRR9097106 (Table 4). The first is a synthetic metagenome of community mixing of *S. pasteurii* with homogenized fecal material, this metagenome is small containing 631.2 Mbp and took approximately 33m using 3 CPUs and buckets of 10Mb. The second metagenome is also synthetic consisting of a co-culture of *Rhodopseudomonas palustris* CGA009

and *Escherichia coli* MG1655 from Indiana University (Bloomington, USA LT1-A25). This metagenome is interesting because besides bigger (counts with more than 2.9 Gbp), it was also obtained of a smaller number of microorganisms, what could be benefit to understand if deepness of sequencing would interfere in the FACS results. The second metagenome took 30h to conclude, when using 3 CPUs and 10Mb buckets. In both tests it was used the Nextera-PE.fa adapters file as standard option, since the adapter sequences not always are available in the metadata.

Table 4. FACS assessment of runs performed with two different metagenomes.

Access	SRR9016022	SRR9097106
Size	631.2 Mbp	2.9 Gbp
User time	33m 49.703s	28h 14m 37.909s
System time	39m 25.515s	30h 57m 35.064s
Real time	1m 17.210s	44m 29.606s
AMP called	129,398	6,376,290

However, tests with different conditions involving, for example, 3 CPUs and buckets of 100Mbp reduced considerably both times of execution to 24 min and 24h, respectively. This means that customizing FACS accordingly to the system available conditions is extremely important to get the best results in the shortest time. Unfortunately, this customization does not follows a rule, but for 100Mbp buckets it was estimated a usage of 10-14 Gbytes of RAM.

The results of FACS runs using both metagenomes are shown in Figures 18 and 19. In both runs using different bucket sizes, there was no differences among the results obtained. Basically, what is observable is the deepness of sequencing seems to be a key-factor to the number of AMPs found in the end of FACS process. However, there is probably due to the higher number of variants available, since FACS performs a clustering using very stringent conditions.

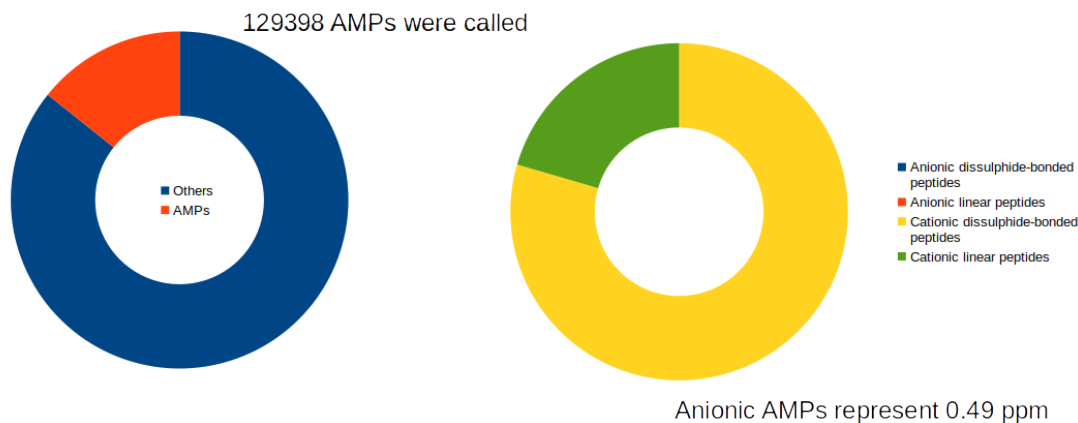


Figure 18. Results of test involving metagenome 631.2Mbp.

An important fact, so far noticed is the low contribution of anionic peptides to the final datasets, what can be a result of the models adopted to filter off the non-AMP sequences. However, it is remarkable the low contribution of this class of peptides even in the previous literature, where most of papers report AMPs as

cationic peptides, neglecting the anionic examples. Other interesting point is the low knowledge available about them. So far, the few informations can be contributing to less representative training datasets that possibly could influence the final prediction models.

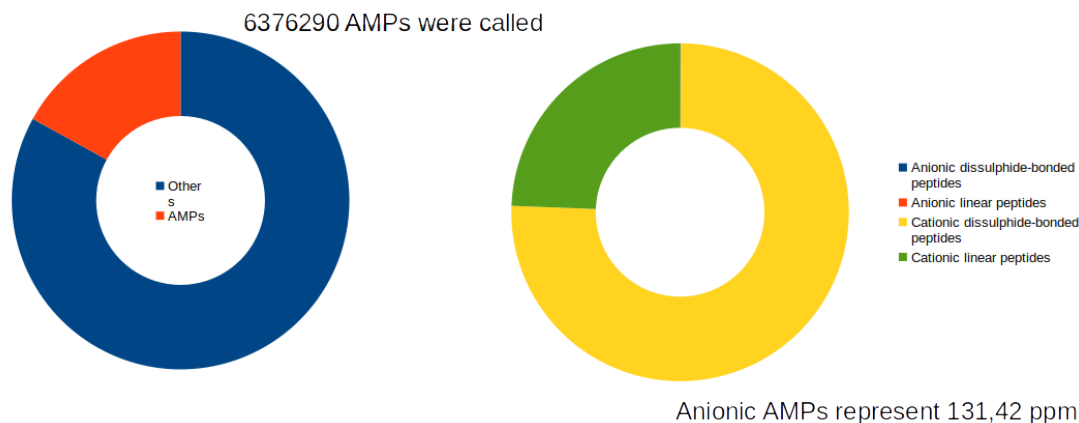


Figure 19. Results of test involving metagenome 2.9Gbp.

In summary, the tests with FACS revealed the potential of this program as a predictor and extractor of sequences. It also was important to show the portability among different studies and the effects of different variables of each study. FACS has shown to be stable and efficient to perform its expected functions in a relative short time in a customizable way.

Applications

FACS can be used in a wide ranging of scenarios, such as: screening for novel AMPs generating candidates to further testing and patenting, as well as, determination of microbiome quorum sensing mechanisms linking AMPs to health conditions or presence of diseases.

Installing

Prior installation make sure your system settings are as follows:

1. Linux (preferably Ubuntu 64 bits, version 18+);
2. You have installed:
 - apt
 - git

- Python Version 3.0 or above
 - Following packages should be already installed in Python environment: sys, os, shutil, scipy, argparse, collections, platform, math, re, numpy (1.13.1), sklearn (0.19.1), matplotlib (2.1.0), pandas (0.20.1).
 - R Version 3.5.2 or above
-

Third party softwares

Also, before start installation make sure you know the needed third party softwares list:

1. To quality trimming of reads and paired-end reads selection and sorting it is used Trimmomatic
 - As complementary to working Trimmomatic needs openjdk-11-jdk-headless.
2. To reads merging it is used pandaseq software.
3. To ORFs prediction it is used ORFm thought to be faster than other ORF prediction systems.
4. To produce descriptors and use the AI models to select peptides, we have used the following R packages:
 - randomForest
 - caret
 - Peptides
 - data.table
 - dplyr
 - parallel
 - doParallel
5. The library FAST from CPANM to speed up perl.
6. The GNUParallel library to speed up the script.
 - Additional libraries needed can include: zlib1g, zlib1g-dev and libpthread-stubs0-dev

7. The pigz software to speed up the compressing and decompressing of files.
 8. The following scripts from the project iLearn to calculate some encodings of the sequences:
 - CTDDClass.py
 - saveCode.py
 - readFasta.py
-

Install procedures

The installation can be performed with downloading the scripts as:

```
$ git clone https://github.com/celiosantosjr/FACS
```

Performing the decompression:

```
$ gunzip FACS-master.gz
```

And executing the installation script:

```
$ sh install.sh
```

====>>>> Be aware that the installation process can require admin privileges.

Usage

Basically, it can be run using the following command line in bash:

```
$ ./FACS.sh [options] --fwd <R1.file.gz> --rev <R2.file.gz>
```

There are few options to make the running of the program a bit customized and speed up process according to the systems settings available.

Basic options:	
-h, -help	Show help page
-fwd	Illumina sequencing file in Fastq format (R1), please leave it compressed and full adress
-rev	Illumina sequencing file in Fastq format (R2), please leave it compressed and full adress
-outfolder	Folder where output will be generated [./]
-outtag	Tag used to name outputs [OUT]
-t, -threads [N]	Number of threads [90% of avaiable threads]
-block	Bucket size (take in mind it is measured in bits and also it determines the memory usage). [100MB]
-adapters	Adapters currently available in Trimmomatic program, if not available you can create one accordingly to its manual. [NexteraPE-PE.fa]
-log	Save results of FACS run to a log file in output folder.

FACS merger

FACS merger was designed to join results of each different metagenome and return a table of abundances of each detected peptide in ppm to each metagenome. Usage:

```
$ ./FACS_merger.sh [options]
```

There are few options to make the running of the program a bit customized and speed up process according to the systems settings available.

Basic options:	
-h, -help	Shows help message
-output [file]	File where output is sent, it needs to be gzipped (ending em .gz)
-t, -threads [N]	Number of threads [90% of avaiable threads]
-reference [folder]	Folder where your reference files are located, if none current folder will be used [Reference_seqs]
