Kernel methods for phenotyping complex plant architecture

Koji Kawamura, Laurence Hibrand-Saint-Oyant, Fabrice Foucher, Tatiana Thouroude, Sébastien Loustau

Kernel methods for phenotyping complex plant architecture

Koji KAWAMURA$^{\star 1,2}$,Laurence HIBRAND-SAINT-OYANT[1], Fabrice FOUCHER[1],Tatiana THOUROUDE[1] and Sébastien LOUSTAU$^{\star 3}$

1. INRA, Institut de Recherche en Horticulture et Semences (INRA, Agrocampus-Ouest, Université d'Angers), SFR 149 QUASAV, 49071 Beaucouzé, FRANCE ,
2. Department of Environmental Engineering, Osaka Institute of Technology, 5-16-1 Ohmiya, Asahi-ku, Osaka, 535-8585, JAPAN,
3. LAREMA, Université d'Angers, 2 Bvd Lavoisier 49045 Angers Cedex, FRANCE
$\star$ have equally collaborated on the paper.

Abstract

The Quantitative Trait Loci (QTL) mapping of plant architecture is a critical step for understanding the genetic determinism of plant architecture. Previous studies adopted simple measurements, such as plant-height, stem-diameter and branching-intensity for QTL mapping of plant architecture. Many of these quantitative traits were generally correlated to each other, which give rise to statistical problem in the detection of QTL. We aim to test the applicability of kernel methods to phenotyping inflorescence architecture and its QTL mapping.

We first test Kernel Principal Component Analysis (KPCA) and Support Vector Machines (SVM) over an artificial dataset of simulated inflorescences with different types of flower distribution, which is coded as a sequence of flower-number per node along a shoot. The ability of discriminating the different inflorescence types by SVM and kernel PCA is illustrated.

We then apply the KPCA representation to the real dataset of rose inflorescence shoots ($n = 1460$) obtained from a 98 *F1*-hybrid mapping population. We find kernel principal components with high heritability (> 0.7), and the QTL analysis identifies a new QTL, which was not detected by a trait-by-trait analysis of simple architectural measurements. The main tools developed in this paper could be use to tackle the general problem of QTL mapping of complex (sequences, 3D structure, graphs) phenotypic traits.

Keywords: Kernel Methods, Inflorescence, QTL mapping, Kernel principal component analysis, Support Vector Machines

1. Introduction

1.1. Motivation

Plant architecture refers to spatial and topological structure of plants ([1]) and determines important aspects of plant function, including productivity ([2]), mechanical stability ([3]), leaf-display efficiency ([4]), and disease resistance ([5]). Therefore, phenotyping method of plant architecture is necessary for (i) understanding the relationship between plant form and function, (ii) the genetic improvement of crop plants, as well as (iii) the development of simulation models of plant growth. Previous studies have thus developed various methodologies for phenotyping plant architectures, such as topological ([6], [7], [8]), three-dimensional ([4], [9]), allometric ([3]), fractal ([10]), and stochastic approach ([11], [12], [13]). These approaches have successfully analyzed and modeled precise plant architecture and its development. However, few studies apply them for phenotyping a large number of plants, which is required in the studies on genetic mapping of Quantitative Trait Loci (QTL) controlling plant architecture.

The QTL mapping of plant architecture is a critical step for understanding the genetic determinism of plant architecture and its genetic improvement by molecular breeding ([2]), but it requires phenotyping a large number of plants ($n > 100$). The elaborated methodologies of phenotyping plant architecture are quite labor-intensive and are not applicable to a large number of plants. Thus, the previous studies on QTL mapping of plant architecture adopted simple geometric and topological measurements of plant architecture, such as plant height, shoot length, diameter, and branching intensity ([14], [15], [16], [8], [17], [18], [19]). Many of these quantitative traits were generally correlated to each other, which give rise to statistical problem in the detection of QTL.

1.2. One-by-one QTL analysis and multiple QTL Mapping

Statistical methods for detecting QTL were originally designed for a trait-by-trait study, mostly using maximum likelihood (see [20]) or linearised approximation (see [21]). Several authors have tried to analyze complex phenotype expression such as architecture of inflorescence. For instance [15] proposes to study maize tassel inflorescence considering 13 correlated inflorescence traits whereas [18] studies a F1 diploid garden rose population with 10 traits associated with the developmental timing and architecture of the inflorescence and with flower production. Then, a one-by-one QTL analysis is proposed to explain the continuous variation of each trait separately. The results of such an approach often suggest that several traits are influenced by the same or linked loci. From the biological viewpoint, many questions involve the interaction between multiple traits and as a result a separate one-by-one analysis is not the most efficient. Moreover, from statistical viewpoint, the power of hypothesis tests (such as QTL mapping) tends to be lower for separate analyses. To answer to this issue, several multiple QTL mapping have been proposed in the last decade, sometimes derived from single trait methods. [22] suggests multiple QTL mapping to combine several traits in an unified analysis. It has the advantage to test a number of biological hypotheses concerning the nature of genetic correlation (pleiotropy, QTL$\times$ environment interaction). This method is shown to be more efficient compared with single trait analysis (see [23]) but suffers from the curse of dimensionality. The number of parameters to estimate is higher and limits statistical power and computing time. Another approach to deal with multiple traits is based on standard multivariate analysis such as principal component analysis. Originally used in [24] for dairy cow data, there are based on a linear combination of the traits in which most of the information is summarised (called the principal component). Then, a single trait analysis can be performed on this first principal component. An attempt is presented also in [15] giving interesting and promising results. The existence of ”PC exclusive QTL” illustrates quiet well the necessity to deal with such a multivariate analysis. [25] proposes extensive simulations to compare such multitraits methods, including another multivariate analysis called discriminant analysis (see [26]).

1.3. Kernel methods

Biology is facing many machine learning challenges. Massive amounts of data are generated, characterized by structured and heteregeneous data (sequences, 3D structures, graphs, networks, SNP) in large quantities and high dimension. At the core of the machine learning methodology, kernel methods have been extensively used to solve many biological problem in the last two decades. We can mention for instance predictive methods for protein function annotation ([27]), gene expression analysis or gene selection for Microarray Data ([28] which surveys the topic of using kernel methods to study biological data). A striking example of a kernel method is the Support Vector Machines (SVM) algorithm due to the pioneering's works of Vladimir Vapnik. The idea of many kernel methods is to map the dataset into an infinite dimensional space, called *feature space*, where the analysis takes place. This mapping is performed by using a so-called kernel function, which measures the similarities between two inputs x and y with the value $k(x, y)$. The construction of various type of kernels, for various type of data, allow to treat many biological problem of pattern recognition, regression estimation or principal component analysis. This idea was originally used for classification with Support Vector Machines (see [29]), or in principal component analysis with Kernel Principal Component Analysis (KPCA, see [30]).

1.4. Our contribution

In this paper, we aim to test the applicability of these kernel methods (namely SVM and KPCA) to QTL mapping of complex plant architectural traits. The main tools developed in this paper could be use to tackle the general problem of QTL mapping of complex (sequences, 3D structure, graphs) phenotypic traits. The idea is to consider these observations directly as inputs and to work in an infinite dimensional feature space thanks to a kernel function. Kernel PCA gives a new and concise representation of the data, which is then used to perform QTL mapping without the problem of multiple, correlated data.

Specifically, we apply the method to the QTL mapping of rose inflorescence architecture (The inflorescence is a flower-bearing branching system, [31]). In the previous work ([18]), QTL mapping of inflorescence architectural traits was performed in a garden rose population. In the population, roses formed a wide variety of inflorescence architecture; a simple inflorescence formed one terminal flower and a few lateral flowers, whereas in a compound inflorescence, lateral shoots continuously branched into higher order shoots and produce numerous flowers (up to 200 flowers). We analyzed total nine traits associated with the length, node number, and branching intensity of inflorescence shoot (see inflorescence architectural traits, Figure 2) and found that most of these nine traits were strongly correlated to each other and they shared QTLs. We finally identified total six common QTLs (cQTLs) as genetic determinants of these nine architectural traits (see cQTL controlling the traits, Figure 2). In the present paper, we use the same rose population and genetic map ([18]) and perform QTL analysis of KPCA scores derived from a simple sequence data of flower distribution along inflorescence shoot. We hypothesize that the KPCA approach identifies a new QTL, which was not detected by the previous work. We first test several kernel functions using artificial data of simulated inflorescences with different types of flower distribution. The capacity of discriminating the different inflorescence types is illustrated by (1) a study of SVM performances of classification using these several kernels (Table 1) and (2) the representation of the artifical dataset using the first three kernel principal coumponents (Figure 3). We then apply kernel PCA to the real dataset of rose inflorescence architecture. The power of distinguishing the genetic variations of inflorescence architecture is evaluated according to the broad-sense heritability of KPCA scores. These KPCA scores are used for QTL analysis.

2. Material and Methods

2.1. Material

2.1.1. A simulated dataset

We first generate a dataset of 6 different artificial inflorescence architectures ('peak', 'asymmetric upper', 'assymetric lower', 'symmetric', 'uniform', 'cyclic') by using R software version2.14.0 (Figure 1). The six inflorescence architecture classes differ in their distribution patterns of flower number per node along main axis. We generated 100 inflorescence shoots with unique rules of determining number of flowers per node for each six architecture classes. Inflorescence architecture is coded by a vector of 8 coordinates, which give the number of flower per node from top (node1) to base (node8) of inflorescence axis.

2.1.2. Real dataset of rose population

Real dataset of inflorescence architecture was collected from the *F1* hybrid population of rose ([18]). This population consists of a progeny of 98 diploid *F1* hybrids from a cross between diploid roses TF x RW. The female parent TF is a commercial cultivar,*The Fairy*, and the male parent RW is a hybrid of *Rosa wichurana*. Both parents develop a highly branched compound inflorescence, and their *F1* hybrids show a large genetic variation of inflorescence architecture.Three replicated clones were created for each 100 genotype (= 98 *F1* hybrid and their parents) by vegetative propagation. A total 300 plants are cultivated in a field of INRA, Angers, France, since 2004. We collected inflorescence data from the 1st order shoot that developed in spring during two years 2008-2009. *Inflorescence* is defined as the top of the 1st order shoot that bore bract-like leaves (*INF1*, Figure 2). For each of the 2nd order shoots that developed from the *INF1*, we count total number of flowers. Then, we define the real dataset as a sequence of flower number per node from the base to tip of the *INF1*. The sum of them corresponds to the total number of flower per inflorescence. Measurements are made on three vigorous shoots per plant in each of the two years, and in total the data of 1460 shoots are obtained and analyze.

2.2. Methods

2.2.1. Kernel Principal Component Analysis (KPCA)

One of the most fundamental steps in data analysis and dimensionality reduction consists of approximating a given data set by a low-dimensional subspace, which is clasically achieved via Principal Component Analysis (PCA). Kernel PCA is the kernelized version of the classical PCA. Given a $(n \times p)$-matrix $X = [X_1 \dots X_p]$ of n observations $x_1, \dots, x_n \in \mathbb{R}^p$, the key step for PCA is the diagonalization of the correlation matrix, given by the inner product $\langle X_i, X_j \rangle$ between variables. Another way of expressing PCA is to consider the diagonalization of the inner product or Graam matrix XX^t, defined as:

$$(XX^t)_{ij} = \langle x_i, x_j \rangle,$$

where $\langle \cdot, \cdot \rangle$ denotes the usual scalar product in the euclidean space $\mathbb{R}^p$. In this case, principal components are calculated from the gram matrix associated to the usual scalar product. Kernel PCA simply mimics this procedure, replacing the inner product matrix by the gram matrix K given by:

$$K_{ij} = k(x_i, x_j),$$

where k is a *kernel function.* For each pair (x_i, x_j), the quantity $k(x_i, x_j)$ measures the similarity between x_i and x_j. From the mathematical viewpoint, k is a symmetric and positive definite function (see [32]). As a result, given a kernel function k, KPCA method provides the best linear combination of the feature variable $k(x_i, \cdot)$, where the dispersion is measured through the kernel function k. In the sequel, we use gaussian kernels $k(x, y) = \exp(-\sigma \|x - y\|^2)$, where $\| \cdot \|$ denotes particular norms.

2.2.2. Specification of the kernels

To apply the KPCA method for phenotyping inflorescence architecture, we develop three simple kernel functions. Consider a vector $x \in \mathbb{R}^p$ as a shoot, where each coordinate corresponds to the number of flowers in a node. In this framework, a kernel is a symmetric and positive definite function which associated to each pairs of shoots (x, y) a similarity measure given by $k(x, y)$.

We first introduce the linear kernel *klin* defined as:

$$\mathrm{klin}(x, y) = \langle x, y \rangle.$$

This kernel is called linear kernel because it corresponds to the usual scalar product in $\mathbb{R}^p$. The KPCA with kernel *klin* is equivalent to the standard PCA.

The gaussian-type kernels are then considered:

$$k(x, y) = \exp(-\sigma \|x - y\|^2), \tag{1}$$

where $\|x - y\|$ stands for a particular distance between x and y and $\sigma > 0$ is a tuning parameter. In the sequel, we called *kdist* the gaussian kernel (1) where $\| \cdot \|$ is the standard euclidean distance in $\mathbb{R}^p$ between shoots:

$$\mathrm{kdist}(x, y) = \exp(-\sigma \|x - y\|^2) = \exp\left(-\sigma \sum_{i=1}^{p} (x_i - y_i)^2\right). \tag{2}$$

We also consider the kernel *kdistderiv*, associated with the discrete derivative of a shoot given by:

$$\begin{aligned} x = (0, 4, 2, 2, 1, 0, 0) \quad \longrightarrow \quad & x' = (4 - 0, 2 - 4, 2 - 2, 1 - 2, 0 - 1, 0 - 0) \\ & x' = (4, -2, 0, -1, -1, 0). \end{aligned}$$

Then, the kernel *kdistderiv* is defined as:

$$kdistderiv(x, y) = \exp(-\sigma \|x' - y'\|^2) = \exp\left(-\sigma \sum_{i=1}^{p-1} ((x_{i+1} - x_i) - (y_{i+1} - y_i))^2\right). \tag{3}$$

This kernel is related to the similarity between variations of flowers between two consecutive node in the architecture.

These gaussian kernels (1)-(3) have a tuning parameter, namely the so-called bandwidth $\sigma > 0$. This parameter has to be chosen in a careful way. We test different σ values from 0.001 to 10. In the sequel, a genetic criterion called heritability is used to select the best value for σ.

2.2.3. Support Vector Machines

The performance of different kernel functions for discriminating different architectures is evaluated by performing Support Vector Machines (SVM).The

SVM is now a standard learning system based on recent advances in statistical learning theory (see [33]). It was originally proposed by Vapnik in [29] to solve the binary classification problem as follows. Consider a learning sample $\{(x_1, y_1), \ldots, (x_n, y_n)\}$, where to each input $x_i \in \mathbb{R}^p$ corresponds a binary response $y_i \in \{-1, +1\}$[1]. Given an input x, it is possible to use a real-valued function $f : \mathbb{R}^p \rightarrow \mathbb{R}$ to assign the class of x: if $f(x) \geq 0$, x is supposed to be in the positive class, and otherwise to the negative class. Linear discrimination (or perceptrons) considers the case where $f(x)$ is a linear function of $x \in \mathbb{R}^p$, so it can be written as:

$$f_{w,b}(x) = \langle w, x \rangle + b, \tag{4}$$

where (w, b) are the parameters that control the decision rule given by $\mathrm{sign}(f_{w,b})$. The idea of SVM is to learn from the learning sample $\{(x_1, y_1), \ldots, (x_n, y_n)\}$ these parameters, by giving an hyperplane which optimally separates the two classes.

In the linear case (4), (w, b) is defined to solve the following optimization problem:

$$\left\{ \begin{array}{l} \min \|w\|^2 + C \sum_{i=1}^{n} \xi_i \\ \text{subject to } y_i(\langle w, x_i \rangle + b) \geq 1 - \xi_i,\ i = 1, \ldots n, \end{array} \right. \tag{5}$$

where $\xi_i \geq 0$ are slack variables and $C > 0$ is a regularization parameter which avoid overfitting. The unique solution of this problem gives the so-called soft margin hyperplane with geometric margin $\gamma = 1/\|w\|^2$ (the distance between the hyperplane and the nearest sample of each class).

Finally, the kernel method of SVM is defined as a soft margin hyperplane in a high dimensional feature space, using a kernel function k. More precisely, deriving the primal Lagrangian for the optimization problem gives rise to the following objective function:

$$W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} y_i y_j \alpha_i \alpha_j (\langle x_i, x_j \rangle + \frac{1}{C} \delta_{ij}),$$

where δ_{ij} is the Kronecker δ defined to be 1 if $i = j$ and 0 otherwise. To move to the more general kernel version, we have to replace in $W(\alpha)$ the scalar

[1] In the sequel, the output of an observations x corresponds to the architecture and belongs to $\{1, 2, 3, 4, 5, 6\}$ (multiclass classification)

product $\langle x_i, x_j \rangle$ by the quantity $k(x_i, x_j)$. The decision rule is given by:

$$\mathrm{sign}(f(x)) = \mathrm{sign}\left(\sum_{i=1}^{n} \alpha_i^* k(x_i, x) + b^*\right).$$

It is equivalent to the hyperplane in the feature space implicitly defined by the kernel k (see [32, Proposition 6.11] for a proof). We refer the interested reader to [32] for a complete and readible introduction about SVM.

In this work, we suggest to use SVM with specific kernels of Section 2.2.2 in order to learn the architectura of simulated shoots ($y \in \{1, 2, 3, 4, 5, 6\}$).We have proceeded as follows:

- Generate 100 observations for each class, i.e. 600 observations.
- Separate these 600 shoots into two samples: a **training set** of 300 shoots (50 per class) and a **test set** of the 300 other shoots.
- Use the training set to construct three SVM classifiers with kernel *kdist-deriv*, *kdist* or *klin*.
- Use the test set to see the classification performances of these three SVM classifiers for different values of σ.

2.2.4. Heritability of Kernel Principal Component of real dataset

We assess the genetic variability of Kernel Principal Components (KPCs) derived from real datasets since QTL analysis assums the high genetic variability of trait. Genetic variability of KPCs is evaluated by calculating its broad-sense heritability. Total variance of KPCs is decomposed into different components using the following model:

$$P_{pqrs} = u + G_p + Y_q + GY_{pq} + R_{(pq)r} + e_{pqrs}$$

where:

- P_{pqrs} is the PCA score calculated for shoot s of plant r in genotype p in year q;
- u is the overall mean;
- G_p is the random effect of genotype p;

- Y_q is the fixed effect of year q;
- GY_{pq} is the random interaction of genotype p and year q;
- $R_{(pq)r}$ is the random effect of replicated plant r nested in genotype p and year q;
- e_{pqrs} is the random residual error for plant r in genotype p in year q.

The total variance (VP) can be decomposed into the variance of genotypic effect (VG), genotype x year interaction (VGY), the variance between replicated plants within the genotype (VR) and residual error variance (VE). The VE includes the variance between replicated shoots within a plant and the error in measurements:

$$VP = VG + VGY + VR + VE.$$

Variance components is estimated based on the restricted maximum likelihood (REML) method. Broad sense heritability (h^2) based on genotypic mean values averaged across years is calculated as follows ([18]):

$$h^2 = \frac{VG}{VG + VGY/L + VR/LM + VE/LMN}$$

where L is the number of replication years, M is the number of replication plants per genotype and N is the number of replication shoots per plant. Average numbers of replications per genotype obtained for each trait is used for the calculation. The analysis is conducted using JMP software version 8.0 (SAS Institute, Inc., Cary, NC).

2.2.5. QTL analysis of Kernel Principal Components (KPCs)

Least square means (LS means) is computed for each KPCs for each genotype. QTL analyses are carried out using MAPQTL 5.0 ([34]) on the LS means and integrated map constructed by [18]. First, we use Kruskal-Wallis test for the rough estimation of QTL location over all KPCs derived from different kernel functions. The test ranks all genotypes according to the LS means, while it classifies them according to their marker genotype. A segregating QTL linked closely to the tested marker will result in large differences in average rank of the marker genotype classes. Based on the genetic map distribution of the significant markers, we estimate the location of underlying QTL. The linkage group with a segregating QTL must reveal a gradient in the test statistic towards the marker with the closest linkage to the QTL.

Secondly, interval mapping is performed for some KPCs in order to confirm the results of Kruskal-Wallis test and to make a more precise estimation of QTL location and effects. A LOD threshold from which a QTL is declared significant is determined according to an error rate of 0.05 over 1000 permutations of the data ([35]). Then, Interval mapping analysis is performed with a step size of 1 cM to find regions with potential QTL effects, i.e., where the LOD score is greater than the threshold. In the region of the potential QTLs, the markers with the highest LOD values are taken as cofactors. A backward elimination procedure is used to select cofactors significantly associated with each trait at $P < 0.02$. Subsequently, multiple QTL mapping (MQM, [36]) is performed with a step size of 1 cM. If LOD scores in the region of the potential QTLs are below the significance threshold, their cofactor loci is removed and MQM mapping is repeated. QTL positions is assigned to local LOD score maxima. Confidence intervals of the map position is indicated in centimorgans corresponding to a 1 or 2-LOD interval. The percentage of phenotypic variance explained by each QTL (r^2) is taken from the MQM mapping output. The total percentage of phenotypic variance explained by all significant QTLs (R^2) is also calculated. The R^2 is then divided by h^2 (percentage scale) to estimate the proportion of genotypic variance explained by the QTL. Allelic effects is also estimated as described in [18].

3. Results

We first apply SVM and KPCA to the simulated dataset. The performances demonstrate that the simulated inflorescence architectures is better classified by a kernel method, in comparison to linear classifier and standard PCA. We then use KPCA for phenotyping real rose inflorescence architecture, in order to perform QTL mapping of inflorescence traits. We find kernel principal components with high heritability (> 0.7). Finally, the QTL analysis identify a new QTL which was not detected by a trait-by-trait analysis of simple architectural measurements.

3.1. SVM and KPCA on simulated dataset

We generate simulated datasets of 6 inflorescence architecture classes and apply SVM method described in Section 2.2.3 to the dataset with three different kernel functions. Table 1 provides the percentage of errors over the second half of observations (the test set) in the procedure of SVM, for three different kernel functions and varying parameter σ. With kernel *kdistderiv* or *kdist*, the error rates is small ($< 13\%$). On the contrary, if we use the linear kernel (which only take the scalar product between two shoots), then we have more than 50% of errors. The minimum error ($< 1\%$) is attained with kernel *kdistderiv* $k(x, y) = \exp(\sigma\|x' - y'\|^2)$ with parameter $\sigma = 0.1$. These results highlight a significant improvment by using kernel method to discriminate architecture inflorescence. Figure 3 shows the first three KPCs with the function *kdistderiv*,$\sigma = 0.05$. Each of the six classes gives a cluster of points, indicating that the method allows to distinguish these six architectures. It is important to stress that a similar representation could be done using the standard PCA. It gives a very bad discrimination between the six architectures. These results validate the representation of shoots with components of these KPCA. It is the motivation to use kernel PCA to look at QTL mapping.

3.2. KPCA on real datasets and QTL Mapping

We then apply the KPCA method on real datasets and assess its performance of phenotyping in terms of the level of heritability. Table 2 (for *klin*) and Table 1S (for *kdist* and *kdistderiv*) summarize the results of calculation of heritability and QTL analysis of KPCs. Generally, the first principal component Z1 has a large heritability compared to the other components, and there are high correlations between Z1 and total number of flowers (*FLW*) per inflorescence (Spearmans rank correlation coefficient > 0.6, p-value < 0.001). This suggests that in the studied population, a large part of genetic variation in inflorescence architecture is owing to the variation in flower number (i.e., size of inflorescence). Kruskal-Wallis test identify the markers that have significantly different Z1 scores between genotype classes. All of them are located in the genomic region of cQTL3 and/or cQTL4, where major QTLs for *FLW* were detected by the previous study (Table 2, Table 1S).

The other principal components of *klin* and *kdist* functions have also substantial genetic variations ($h^2 > 0.5$). Kruskal-Wallis tests for these components show that most of their significant markers are located in the six cQTL regions (Table 2, 1S). These loci were previously detected by QTL mapping of inflorescence architectural traits, such as the length (*LF*), the node number (*NF*) and the branching intensity (*BIF*) of inflorescence shoots (Figure 2). This indicates that our kernel principal components derived from the data of a sequence of flower number along inflorescence shoot can integrate the architectural variations of inflorescence shoots. In contrast to the *klin* and *kdist* function, the *kdistderiv* function did not work well. About half of principal components extracted by *kdistderiv* function had no significant genetic variations (NS, Table1S). Kruskal-Wallis test detected only a few significant markers, all of which were located within the cQTL regions (Table 1S).

Interestingly, Kruskal-Wallis test for the KPCs of *kdist* function with $\sigma = 0.025$ and 0.05 detects significant markers in linkage group 6 (Table 1S), where the previous study have not detected any QTLs for inflorescence architecture. This indicates the discover of a new QTL. In order to confirm the result, we perform MQM mapping analysis on the KPCs derived from the function *kdist* with $\sigma = 0.025$ (Table 3).

MQM mappings for first five components derived from *kdist* function with $\sigma = 0.025$ identify total 11 QTLs, most of which have overlapping confidence intervals with known cQTL regions (Figure 4), as shown by Kruskal-Wallis tests (Table 1S). An exception is the QTL *Z2dist-1* in LG 6, where the previous study did not detect any QTLs. A genetic marker, *RoTFL1b*, a homologue gene of *TERMINAL FLOWER 1* of *Arabidopsis thaliana* ([37]), shows significant differences in the Z2 score of *kdist* function with $\sigma = 0.025$ and 0.05 (Kruskal-Wallis test, $P < 0.001$), indicating the presence of a new QTL in this region. The genotypic correlation analysis shows that the Z2 score of *kdist* (named as *Z2kdist*) is not significantly correlated with any architectural traits, such as the internode length (*LF*), the node number (*NF*) or the branching intensity (*BIF*) of inflorescence axes (Spearmans rank correlation test, $P > 0.2$). Moreoevr, it is weakly correlated with the total number of flower (*FLW*) per inflorescence (ρ_{S} = -0.23, $P > 0.05$). Thus, the *Z2kdist* is not characterized by simple architectural traits, such as the length, number, and branching intensity of nodes. It is also not a simple measure of inflorescence size. As a result, The newly identified QTL *Z2kdist-1* might be involved with the control of flower distribution along inflorescence axis.

4. Discussion

4.1. SVM and KPCA on simulated dataset

In this paper, we demonstrate a low error of classification by Support Vector Machines (SVM) when the six simulated architectural classes are analyzed by the kernel *kdistderiv* and *kdist* (Table 1). The six classes are well separated in a three dimensional representation (Figure 3) by the kernel version of Principal Component Analysis (KPCA). Thus, the kernels *kdist* and *kdistderiv* allow us to distinguish symmetric, dyssimetric, cyclic, and constant distribution patterns precisely. In contrast, the linear kernel, which corresponds to standard PCA, does not show a good performance of classification (Table 1). This indicates the significance to use a kernel method instead of classical PCA method.

Our KPCA method for phenotyping plant architecture can be applicable to a wide range of botanical and ecological datasets. For example, the number of leaflet per leaf produced on a rose shoot (e.g., 3,3,3, 5, 7, 9, 9, 7, 5, 3, ...) and the length of lateral shoot produced by a main shoot ([38]) can be characterized by KPCA. Phenological data such as the production of flower per plant during a period (e.g., 0, 0, 5, 10, 15, 20, 10, 3, 0, 1, 0, 0, ...) can be also characterized by KPCA. As shown by simulated dataset (Fig. 3), our KPCA can distinguish the cyclic-type phenology from constant and monotonic types.

Another possible direction is to use different kernel functions, such as kernels on graphs (see [39]). Kernels on graphs, such as the popular Laplacian kernel, can deal with a detailed topological structure of the shoots, presented as a graph. It will allow us to analyze deeply the real architecture of plants and more complex datasets.

4.2. KPCA on real datasets and QTL Mapping

This contribution demonstrates the applicability of KPCA method for QTL mapping of a complex plant architectural trait, namely the inflorescence architecture. We assess the usefulness of different kernel functions based on the calculation of heritability of KPCA components. This allows us to select the kernel function that discriminates well the genetic variance of the focused traits in the studied population. The QTL analysis of kernel principal components identifies a new QTL, which was not detected by a trait-by-trait analysis.

[15] applied Principal Component (PC) analysis on maize tassel inflorescence architecture and identified the PC exclusive QTL that might be involved in regulation of multiple traits, which could not be detected using trait-by-trait analysis. In the present paper, the KPCs with *klin* function, which is the same procedure of standard PCA, does not detect any new QTLs (Table 2), whereas our KPCs with *kdist* function (with = 0.025) identified a new QTL. This indicates that our KPCA is potentially more powerful tool

for phenotyping plant architecture compared to the standard PCA. However, the KPCA with *kdistderiv* function does not provide any KPCs with high heritability and new QTLs (Table 1S). A large part of phenotypic variations extracted by *kdistderiv* function is therefore not associated with the genetic variations but is due to between- and within-plant variations of inflorescence architecture. Thus, the success of the extraction and the integration of phenotypic data for QTL analysis depend on the choice of a kernel function.

Our kernel approach identified a new QTL, *Z2kdist-1*, which could not be detected by a trait-by-trait analysis. We have tried to characterize the function of *Z2kdist-1*. We can conjecture that the *Z2kdist* represents the distribution pattern of flower along inflorescence axis. To test the hypothesis, we can examine the correlation between the *Z2kdist* and simple indices of flower distribution along inflorescence axis. The simple indices of flower distribution are obtained by coutning the number of nodes where the accumulative number of flower attains 50 percent of total number of flower. The calculations are done for each shoot both from the base and the tip of inflorescence axis (*INF1*, Fig. 2). The indices, obtained by counting from the base and the tip, are named as *B50* and *T50*, respectively. Either the *B50* or the *T50* are not significantly correlated with the *Z2kdist* ($P > 0.1$). QTL analysis does not detect significant QTLs on LG6 either for the *B50* or for the *T50* (data not shown). Therefore, the *Z2kdist* could not be characterized by the simple indices of flower distribution tested here. A detailed pattern analysis (e.g., [11]) may be necessary to interpret the *Z2kdist* and the function of *Z2kdist-1*.

The *RoTFL1b* is a candidate gene for the *Z2kdist-1*. In *Arabidopsis thaliana*, *TFL1* is expressed in shoot apical meristem to maintain meristem indeterminacy and control inflorescence architecture ([40]). Overexpression of *TFL1* delays flower formation and forms a highly branched inflorescence, while *tfl1* mutants have a short vegetative phase and form a simple determinate inflorescence with a terminal flower ([41]). The structure and function of *TFL1* gene is greatly conserved in plants (reviewed by [42]). We recently demonstrated that *RoKSN*, another *TFL1* member in rose, is expressed in shoot apical meristem and plays a role in the repression of flowering, and *ksn* mutants have a continuous flowering habit ([37]). Given the high degree of sequence similarity between *RoKSN* and *RoTFL1b* ([37]), it is likely that the *RoTFL1b* is also involved in the control of floral transition and inflorescence development in rose. Future expression analysis and physiological study will be necessary to clarify the hypothesis.

5. Acknowledgements

This work was supported by grants from the Département de Génétique et d'Amélioration des Plantes INRA, Région Pays de la Loire, and JSPS KAKENHI Grant Number 24688004. The authors gratefully acknowledge Stéphanie Gaillard for correcting the English.

References

[1] D. Barthlmy, Y. Caraglio, Plant architecture: a dynamic, multilevel and comprehensive approach to plant form, structure and ontogeny, Ann. Bot. 99 (2007) 375–407.

[2] T. Sakamoto, M. Matsuoka, Generating high-yielding varieties by genetic manipulation of plant architecture, Current Opinion in Biothechnology. 15 (2004) 144–147.

[3] K. J. Niklas, Plant allometry: The scaling of form and process, University of Chicago Press, 1994.

[4] R. W. Pearcy, H. Muraoka, F. Valladares, Crown architecture in sun and shade environments: assessing function and trade-offs with a three-dimensional simulation model, New Phytol. 166 (2005) 791–800.

[5] C. A., D. Milbourne, L. Ramsay, R. Meyer, C. Chatot-Balandras, P. Oberhagemann, W. D. Jong, C. Gebhardt, E. Bonnel, R. Waugh, Qtl for field resistance to late blight in potato are strongly correlated with maturity and vigour, Molecular breeding 5 (1999) 387–398.

[6] C. Godin, Y. Caraglio, A multiscale model of plant topological structures, J. Theor. Biol. 191 (1998) 1–46.

[7] P. Ferraro, C. Godin, A distance measure between plant architectures, Ann. For. Sci. 57 (2000) 445–461.

[8] V. Segura, C.-E. Durel, E. Costes, Dissecting apple tree architecture into genetic, ontogenetic and environmental effects: Qtl mapping, Tree Genetics and Genomes 5 (2009) 165–179.

[9] C. Godin, E. Costes, H. Sinoquet, A method for describing plant architecture which integrates topology and geometry, Ann. Bot. 84 (1999) 343–357.

[10] P. Ferraro, C. Godin, P. Prusinkiewicz, Toward a quantification of self-similarity in plants, Fractals 13 (2005) 91–109.

[11] Y. Gudon, D. Barthlmy, Y. Caraglio, E. Costes, Pattern analysis in branching and axillary flowering sequences, J. Theor. Biol. 212 (2001) 481–520.

[12] E. Costes, Y. Gudon, Modelling branching patterns on 1-year- old trunks of six apple cultivars, Ann. Bot. 89 (2002) 513–524.

[13] M. Renton, Y. Gudon, C. Godin, E. Costes, Similarities and gradients in growth unit branching patterns during ontogeny in fuji apple trees: a stochastic approach, J. Exp. Bot. 57 (2006) 3131–3143.

[14] V. Segura, C. Cilas, F. Laurens, E. Costes, Phenotyping progenies for complex architectural traits: a strategy for 1-year-old apple trees (malus x domestica borkh.), Tree Genetics and Genomes 2 (2006) 140–151.

[15] N. Upadyayula, J. Wassom, M. Bohn, T. Rocheford, Quantitative trait loci analysis of phenotypic traits and principal components of maize tassel inflorescence architecture, Theor Appl Genet 113 (2006) 1395–1407.

[16] K. Onishi, Y. Horiuchi, N. Ishigoh-Oka, K. Takagi, N. Ichikawa, M. Maruoka, Y. Sano, A qtl cluster for plant architecture and its ecological significance in asian wild rice, Breed. Sci. 57 (2007) 7–16.

[17] X. Song, T. Zhang, Quantitative trait loci controlling plant architectural traits in cotton, Plant Sci. 177 (2009) 317–323.

[18] K. Kawamura, L. H.-S. Oyant, L. Crespel, T. Thouroude, D. Lalanne, F. Foucher, Quantitative trait loci for flowering time and inflorescence architecture in rose, Theor Appl Genet 122 (4) (2011) 661–675.

[19] F. Zhang, J. Jiang, S. Chen, F. Chen, W. Fang, Mapping single-locus and epistatic quantitative trait loci for plant architectural traits in chrysanthemum, Mol. Breed. 30 (2012) 1027–1036.

[20] E. Lander, D. Botstein, Mapping mendelian factors underlying quantitative traits using rflp linkage maps, Genetics 121 (1989) 185–199.

[21] C. Haley, S. Knott, A simple regression method for mapping quantitative trait loci in line crosses using flanking markers, Heredity 69 (1992) 315–324.

[22] C. Jiang, Z. Zeng, Multiple trait analysis of genetic mapping for quantitative trait loci, Genetics 140 (1995) 1111–1127.

[23] C. Hackett, R. Meyer, W. Thomas, Multi-trait qtl mapping in barley using multivariate regression, Genet. Res. Camb. 77 (2001) 95–106.

[24] J. Weller, G. Wiggans, P. Vanraden, M. Ron, Application of canonical a transformation to detection of quantitative trait loci with the aid of genetic markers in a multi-trait experiment, Theor Appl Genet 92 (1996) 998–1002.

[25] H. Gibert, P. LeRoy, Comparison of three multitrait methods for qtl detection, Genet. Sel. Evol. 35 (2003) 281–304.

[26] K. Mardia, J. Kent, J. Bibby, Discriminant analysis, Multivariate Analysis, Academic Press, London (1979) 300–332.

[27] X. Zhou, C. Chen, Z. Li, X. Zhou, Using chou's amphilic pseudo-amino acid composition and support vector machine for prediction of enzyme subfamily classes, J Theor Biol 248 (2007) 546–551.

[28] B. Scholkopf, K. Tsuda, J.-P. Vert, Kernel methods for computational biology, MIT, 2007.

[29] B. Boser, I. Guyon, V. Vapnik, A training algorithm for optimal margin classifiers, in: Computational Learning Theory, pp. 144–152.

[30] B. Schölkopf, A. Smola, Learning with Kernels, MIT Press, 2002.

[31] D. F. Weberling, Morphology of flowers and inflorescences, Cambridge University Press, 1992.

[32] N. Cristianini, J. Shawe-Taylor, An introduction to Support Vector Machines and other kernel based learning methods, Cambridge University Press, 2000.

[33] V. Vapnik, Statistical Learning Theory, John Wiley Sons, 1998.

[34] J. W. V. Ooijen, MAPQTL 5.0 software for the mapping of quantitative trait loci in experimental populations, Plant Research International, Wageningen, 2004.

[35] G. Churchill, R. Doerge, Empirical threshold values for quantitative trait mapping, Genetics 138 (1994) 963–971.

[36] R. Jansen, P. Stam, High resolution of quantitative traits into multiple loci via interval mapping, Genetics 136 (1994) 1447–1455.

[37] H. Iwata, A. Gaston, A. Remay, T. Thouroude, J. Jeauffre, K. Kawamura, L. H.-S. Oyant, T. Araki, B. Denoyes, F. Foucher, The tfl1 homologue ksn is a regulator of continuous flowering in rose and strawberry, Plant J. 69 (2012) 116–125.

[38] P. Morel, G. Galopin, N. Dons, Using architectural analysis to compare the shape of two hybrid tea rose genotypes, Sci.Hort. 120 (2009) 391–398.

[39] K. Tsuda, J.-P. Vert, Kernel Methods in Computational Biology, MIT Press, 2004.

[40] P. Prusinkiewicz, Y. Erasmus, B. Lane, L. D. Harder, E. Coen, Evolution and development of inflorescence architectures, Science 316 (2007) 1452–1456.

[41] D. Bradley, O. Ratcliffe, C. Vincent, R. Carpenter, E. Coen, Inflorescence commitment and architecture in arabidopsis, Science 275 (1997) 80–83.

[42] R. C. McGarry, B. G. Ayre, Manipulating plant architecture with members of the cets gene family, Plant Sci. 188-189 (2012) 71–81.

Figure 1

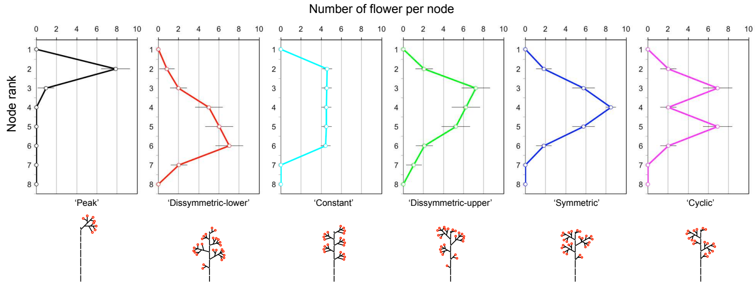
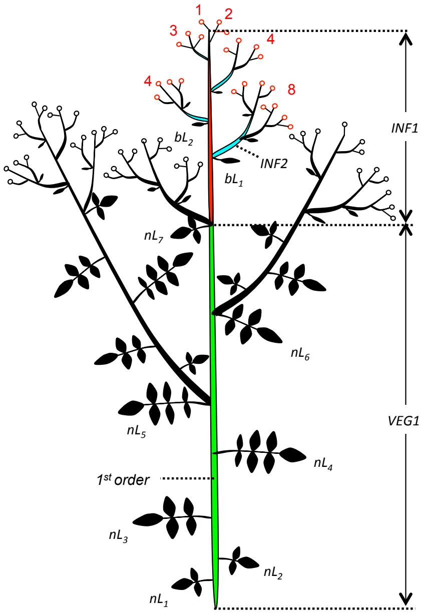


Figure 1. Graphical presentation of six simulated inflorescences. Vertical axis shows node ranks along an inflorescence axis from top (node 1) to base (node 8). Horizontal axis shows the number of flower per node (f_x). Mean value (circle) of randomly generated 100 inflorescences was shown with standard deviation (bar). In each of the six classes, there is a unique rule of determining number of flower per node. *Peak* class: $f_1 = 0$, $f_2 =$ Random Integer (RI) from 6-11, $f_3 =$ RI from 0-3, f_4, , $f_8 = 0$; *Dissymmetric-lower* class: $f_1 = 0$, $f_2 =$ RI from 0-3, $f_3 =$ RI from 1-4, $f_4 = f_5$-1, $f_5 = f_6$-1, $f_6 =$ RI from 5-10, $f_7 =$ RI from 1-4, $f_8 = 0$; *Constant* class: $f_1 = 0$, f_2, , $f_6 =$ RI from 4-6, $f_7 = f_8 = 0$; *Dissymmetric-upper* class: $f_1 = 0$, $f_2 =$ RI from 1-4, $f_3 =$ RI from 5-10, $f_4 = f_3$-1, $f_5 = f_4$-1, $f_6 =$ RI from 1-4, $f_7 =$ RI from 0-3, $f_8 = 0$; *Symmetric* class: $f_1 = 0$, $f_2 =$ RI from1-4, $f_3 =$ RI from 4-8, $f_4 =$ RI from 8-10, $f_5 = f_3$, $f_6 = f_2$, $f_7 = f_8 = 0$; *Cyclic* class: $f_1 = 0$, $f_2 =$ RI from 1-4, $f_3 =$ RI from 5-10, $f_4 = f_2$, $f_5 = f_3$, $f_6 = f_2$, $f_7 = f_8 = 0$. Generation of RI was made by R version 2.14.0.

Figure 2



Definition

nL	Normal leaf with at least 3 leaflets
bL	Bract-like leaf with 1-2 small inmature leaflets
VEG1	Vegetative part of 1st order shoot with *nLs*
INF1	Inflorescence part of 1st order shoot with *bLs*
INF2	The longest 2nd order shoot which develops from *INF1*

Inflorescence architectural traits

NV1	Number of nodes on *VEG1*
NF1	Number of nodes on *INF1*
NF2	Number of nodes on *INF2*
LV1	Average length of internodes of *VEG1* (= Length of *VEG1* / *NV1*)
LF1	Average length of internodes of *INF1* (= Length of *INF1* / *NF1*)
LF2	Average length of internodes of *INF2* (= Length of *INF2* / *NF2*)
NBF2	Number of 3rd order shoots on *INF2*
BIF2	% of lateral meristems, which develop into 3rd order shoots (= 100**NBF2* / *NF2*)
FLW	Total number of flower produced by *INF1*

Common QTL (cQTL) controlling the traits

NV1	*cQTL1,*	*cQTL2,*	*cQTL3,*		*cQTL5*	
NF1	*cQTL1,*	*cQTL2,*	*cQTL3*			
NF2	*cQTL1,*		*cQTL3*			
LV1	*cQTL1,*			*cQTL4,*	*cQTL5,*	*cQTL7*
LF1			*cQTL3,*	*cQTL4,*	*cQTL5*	
LF2			*cQTL3,*	*cQTL4,*	*cQTL5*	
NBF2	*cQTL1,*		*cQTL3,*	*cQTL4,*	*cQTL5*	
BIF2			*cQTL3,*	*cQTL4*		
FLW	*cQTL1,*		*cQTL3,*	*cQTL4*		

Figure 2. Pictorial representation of branching structure of 1st order shoot and inflorescence garden rose. Definitions of terms are on the right. Open circle indicates a flower. The main axis corresponds to the 1st order shoot, and the lateral shoots developing from the 1st order axis are 2nd order shoots. The boundary between vegetative part (*VEG1*) and the inflorescence (*INF1*) of the 1st order shoot is defined according to the changes in leaf morphology from normal leaves (*nLs*) to bract-like leaves (*bLs*). The numbers of flower produced by 2nd order shoots are counted along *INF1* axis from the base (8, 4, 4, 3, 2, 1) and are analysed by kernel method as a vector. Other architectural trait values of the picture are as follows; *NV1* = 7, *NF1* = 6, *NF2* = 4, *NBF2* = 3, *BIF2* = 75, *FLW* = 22. Common QTL regions (cQTL) controlling these traits are also listed on the right. After modification of Figure 1 from [18].

Figure 3

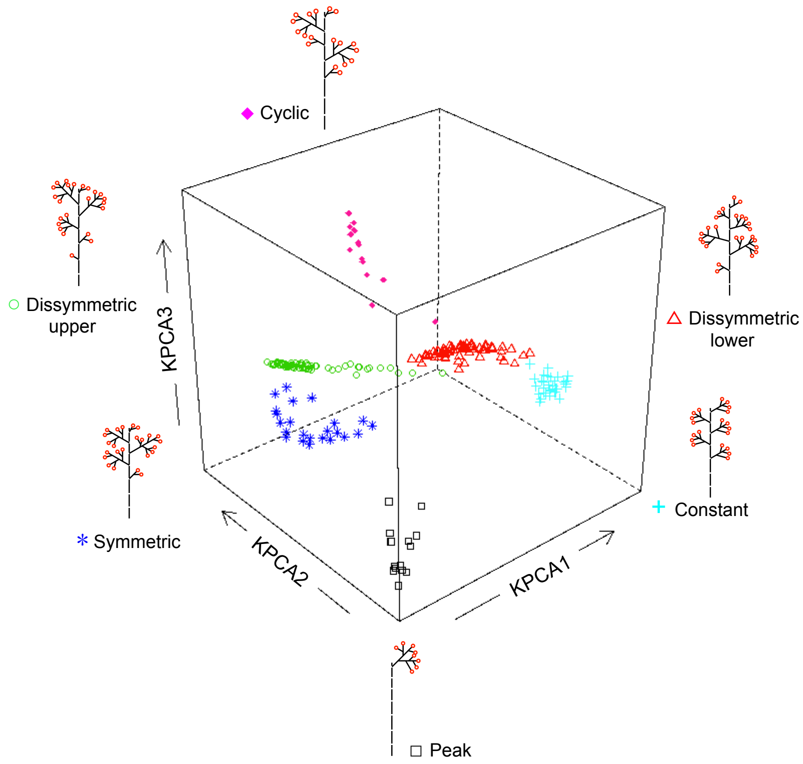


Figure 3. Three-dimensional scatter plot of first three components of the kernel principal component analysis with the kernel function *kdistderiv*, = 0.05 on simulated dataset. Each of the six classes gives a cluster of points, indicating that the method allows to distinguish these six architectures.

Figure 4

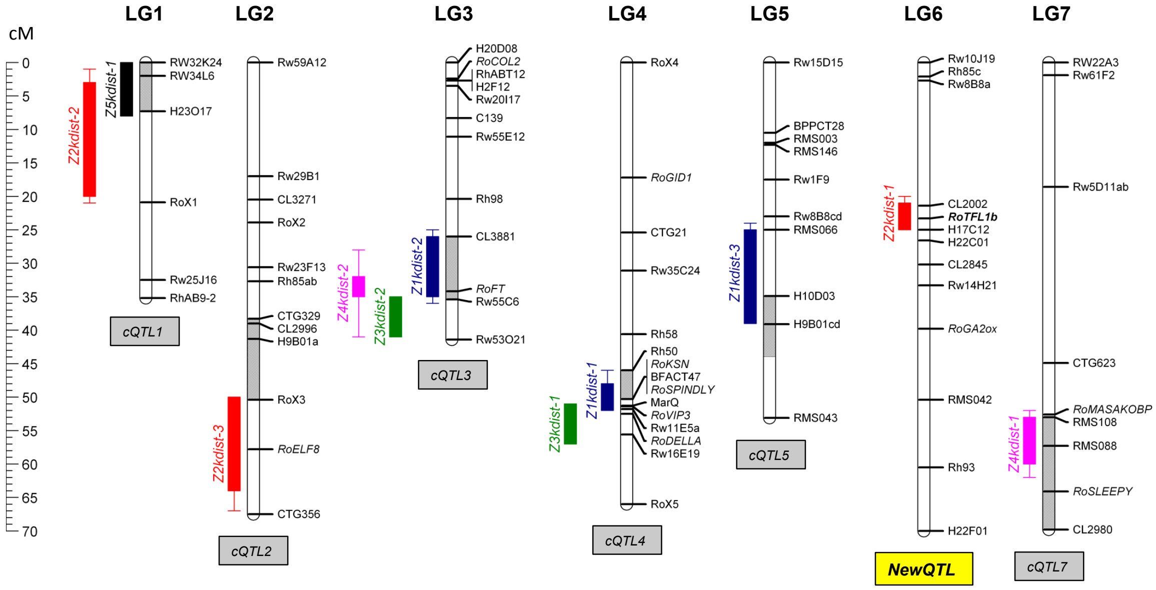


Figure 4. Genetic map locations of QTLs for five kernel principal components (KPCs) detected by multiple QTL mapping in 98 *F1* diploid roses derived from the cross TF x RW. The KPCs are obtained by applying kernel function *kdist* with = 0.025 to the sequence data of flower number per node along inflorescence axes. Common genomic regions of QTLs (cQTLs) for 10 inflorescence developmental traits previously identified by [18] are also indicated. QTLs are illustrated by boxes whose length represents the LOD-1 confidence interval. Extended lines represent the LOD-2 confidence interval. Left bar shows the map scale in cM. For QTL abbreviation, see Table 3.

Table1. The percentages of classification errors of six simulated architectural classes during the procedure of Support Vector Machine (SVM), using different kernel function and σ values.

	Percentage of wrong classification (%) ± SE				
Kernel function	σ = 0.001	σ = 0.01	σ = 0.1	σ = 1	σ = 10
kdistderiv	9.38±1.6	5.3±1.3	**0.35±0.4**	2.5±2.2	6.7±4.1
kdist	12.82±2.3	10.6±1.8	2.3±1.3	**1.2±1.3**	7.3±4.7
klin	56.5±8.3	56.5±8.3	**56.5±8.3**	56.5±8.3	56.5±8.3

See the *Methods* section fot the definition of simulated architectural class (Section *2.1.1*) and kernel function (Section *2.2.1*)*,* and for the procedure of SVM (Section *2.2.3*).

Table2. Heritability and the estimation of QTL locations by Kruskal-Wallis test for kernel principal components (KPCs) derived from the sequence data of flower number per node along inflorescence axes in 98 *F1*-hybrid roses (TF x RW) using the kernel function *klin*.

KPC[a]	Heritability[b]	QTL location and significance level[c]
Z1klin	0.93	cQTL3****, cQTL4****
Z2klin	0.81	ND
Z3klin	0.66	cQTL2***, cQTL7**
Z4klin	0.77	cQTL1****
Z5klin	0.72	cQTL1***, cQTL4**, cQTL7*

[a], Only first 5 principal components (Z1, Z2, …, Z5) were analyzed.

[b], Broad-sense heritability was estimated for each principal components (*See* section *2.2.4* for detailed calculation methods).

[c], Genetic map locations of markers that had significantly different *Z* scores between genotype classes (*P*<0.005, Kruskal-Wallis test). cQTL, a common QTL region where the QTLs for inflorescence architectures detected (Fig. 2); Number indicates the linkage group; The highest significance level of the markers located in the regions: **P*<0.005, ***P*<0.001, ****P*<0.0005, *****P*<0.0001. ND. No significant markers detected (*P*>0.005).

Table 3. QTLs for kernel principal components (KPCs) derived from the sequence data of flower number per node along inflorescence axes in 98 *F1*-hybrid roses (TF x RW) using the kernel function *kdist* with σ = 0.025, detected by MQM mapping procedure.

			QTL position				Allelic effect[d]					% PVE[e]		
KPC	QTL	LOD[a]	LG	cM	Confidence interval[b]	Cofactor[c]	*Af*	*Am*	*D*	(*Af*/*Am*)	(*D*/*A*)	r^2	R^2	(R^2/h^2)[f]
Z1kdist	*Z1kdist-1*	12.32 (3.8)	LG4	50	48-52 (46-52)	*RoKSN*	-0.08	2.25	-0.11	0.03	0.10	28.5		
	Z1kdist-2	9.52 (3.8)	LG3	34	26-35 (25-36)	*RoFT*	-0.62	1.73	0.67	0.36	0.57	20.5		
	Z1kdist-3	4.22 (3.8)	LG5	36	25-39 (24-39)	H10_D03	0.73	-1.00	0.00	0.73	0.00	8.5		
													57.5	0.65
Z2kdist	*Z2kdist-1*	5.72 (3.8)	LG6	23	21-25 (20-25)	*RoTFL1b*	-0.19	-1.95	-0.08	0.10	0.07	22.5		
	Z2kdist-2	3.72 (2.7)	LG1	10	3-20 (1-21)	H23_O17	0.85	1.50	0.37	0.57	0.32	17.6		
	Z2kdist-3	3.43 (2.9)	LG2	59	50-64 (50-67)	*RoELF8*	0.15	-0.75	1.43	0.20	3.19	17.1		
													57.2	0.79
Z3kdist	*Z3kdist-1*	3.88 (3.8)	LG4	56	51-57 (51-57)	Rw16E19	0.38	-1.35	0.79	0.28	0.91	14.7		
	Z3kdist-2	3.43 (2.7)	LG3	41	35-41 (35-41)	Rw53O21	0.29	-1.17	-0.85	0.25	1.16	12.9		
													27.6	0.43
Z4kdist	*Z4kdist-1*	7.26 (3.8)	LG7	57	53-60 (52-62)	RMS088	2.02	-0.41	0.23	4.92	0.19	26.6		
	Z4kdist-2	2.66 (2.6)	LG3	34	32-35 (28-41)	*RoFT*	-0.74	-0.66	-0.82	1.11	1.17	8.7		
													35.3	0.49
Z5kdist	*Z5kdist-1*	5.63 (3.8)	LG1	5	0-8 (0-8)	RW34L6	-1.33	-1.41	0.72	0.94	0.53	24.4		
													24.4	0.33

[a] Maximum LOD socre with threthfold LOD in parenthesis; 3.8 = genome-wide level of significance, 2.6-2.9 = chromosome-wide level of significance ($P < 0.05$).

[b] 1-LOD interval cM with 2-LOD interval cM in parenthesis

[c] Markers used as cofactor for MQM mapping procedure

[d] Allelic effect calculated based on estimated phenotypic value, u_{ac}, u_{ad}, u_{bc}, u_{bd} associated to each of the 4 possible genotypic classes, *ac*, *bc*, *ad*, and *bd*, deriving from the cross ab (female) x cd (male). Af is female additivity calcuated as [(uac – ubc) + (uad – ubd)]/(2*SD), Am is male additivity caculated as [(uac – uad) + (ubc – ubd)]/(2*SD), and D is the overall dominance effect calculated as D =[(uac + ubd) – (uad + ubc)]/(2*SD). Values are standardized by dividing standard deviation (SD) of the trait. (Af/Am) is relative effect of female / male additivity calculated as ┃Af┃/┃Am┃ , and (D/A) is relative effect of dominance / additivity calculated as ┃2*D┃ / (┃Af┃ +┃Am┃).

[e] Percentage of phenotypic variance explained by each QTL (r^2) and by all significant QTLs (R^2) in each trait.

[f] Proportion of genetic variance explained by QTL calculated by dividing R^2 by broad-sense heritability (h^2, %).

Table1S. Heritability and the estimation of QTL location by Kruskal-Wallis test for kernel principal components derived from the sequence data of flower number per node along inflorescence axes in 98 *F1*-hybrid roses (TF x RW) using the kernel function *kdist* or *kdistderiv* and σ values.

		kenerl function = *kdist*		kenerl function = *kdistderiv*	
σ	*Z*[a]	Heritability[b]	QTL location and significance level[c]	Heritability[b]	QTL location and significance level[c]
0.01	Z1	0.89	cQTL3****, cQTL4****	0.91	cQTL3****, cQTL4****
	Z2	0.78	ND	0.43	ND
	Z3	0.71	ND	***NS***	ND
	Z4	0.74	cQTL7***	***NS***	ND
	Z5	0.63	ND	***NS***	ND
0.025	Z1	0.88	cQTL3****, cQTL4****, cQTL5*	0.90	cQTL3****, cQTL4****
	Z2	0.73	***NewQTL6*****	0.36	ND
	Z3	0.64	cQTL4*	0.42	ND
	Z4	0.71	cQTL7***	***NS***	ND
	Z5	0.75	cQTL1***	***NS***	ND
0.05	Z1	0.87	cQTL3****, cQTL4****	0.89	cQTL3****, cQTL4****
	Z2	0.69	***NewQTL6*****	***NS***	ND
	Z3	0.62	cQTL4**, cQTL2****	0.55	ND
	Z4	0.72	cQTL7****	***NS***	ND
	Z5	0.51	ND	***NS***	ND
0.1	Z1	0.84	cQTL4****	0.87	cQTL3****, cQTL4****
	Z2	0.71	cQTL4**	***NS***	ND
	Z3	0.55	cQTL4**, cQTL3****	0.39	ND
	Z4	0.73	cQTL7****, cQTL4****	0.55	ND
	Z5	0.64	ND	***NS***	ND
0.25	Z1	0.81	cQTL4***	0.83	cQTL4****
	Z2	0.75	cQTL4****, cQTL3**	***NS***	cQTL5**
	Z3	0.66	cQTL4****, cQTL1**	0.65	ND
	Z4	0.84	ND	***NS***	ND
	Z5	0.56	***NewQTL6*****	***NS***	ND
0.5	Z1	0.76	cQTL4****	0.81	cQTL4****
	Z2	0.74	ND	***NS***	cQTL4***
	Z3	0.54	cQTL4***	***NS***	ND
	Z4	0.39	ND	0.52	cQTL5****
	Z5	0.81	cQTL4***, cQTL3***	0.52	ND

[a], Only first 5 principal components (Z1, Z2, …, Z5) were analyzed. Because σ ≥1 did not produce Z scores with high heritability (>0.65), only the results with σ < 1 are shown.

[b], Broad-sense heritability was estimated for each principal components (*See* section *2.2.4* for detailed calculation methods); ***NS***, Not Significantly different from zero (*P*>0.05).

[c], Genetic map locations of markers that had significantly different *Z* scores between genotype classes (*P*<0.005, Kruskal-Wallis test). cQTL, a common QTL region where the QTLs for inflorescence architectures detected (Fig. 2); ***NewQTL***, a newly identified QTL region; Number indicates the linkage groups; The highest significance level of the markers located in the regions: **P*<0.005, ***P*<0.001, ****P*<0.0005, *****P*<0.0001. ND. No significant markers detected (*P*>0.005).